FROM ERROR PROPAGATION TO RECOVERY: PRINCIPLES FOR PERSISTENT MEMORY IN MULTI-AGENT SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Error propagation undermines reliable decision-making in large language model-based multi-agent systems (LLM-MAS). Existing work uses persistent memory to mitigate this problem, but lacks a theoretical characterization of how memory enables recovery. This leaves memory design without theoretical guidance on *what to remember* and *when to intervene*, potentially leading to unnecessary memory use and increased *token and storage costs*. To address this gap, we connect error propagation and recovery analysis, establishing the necessity of persistent memory in LLM-MAS under information loss and deriving principles for memory design. Specifically, we derive a closed-loop error propagation law coupling agent communication and environment feedback. We further analyze error recovery under information loss and show that persistent memory reduces the optimal error from random guessing to zero. Guided by this theoretical analysis, we develop RECOVER, an in-task memory for efficient error recovery. Experiments across five benchmarks and three MAS frameworks demonstrate performance gains over in-task and cross-task memory baselines, with efficiency analyses showing reduced token usage and storage overhead. [1]

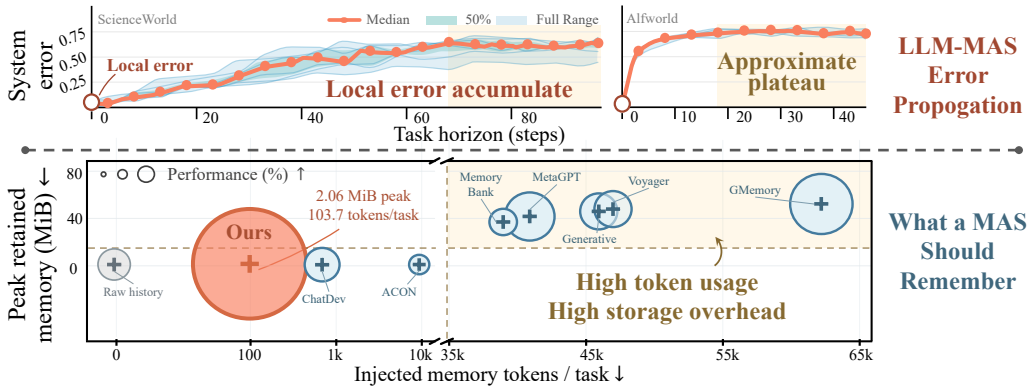

Figure 1: **Error propagation and memory in LLM-MAS.** *Top.* Localized errors propagate in LLM-MAS, accumulating into persistent system-level deviations supported by both theory and experiments. *Bottom.* Our theory-driven in-task memory selectively retains compact task-sufficient states, improving task performance with low token usage and storage overhead.

1 INTRODUCTION

Error propagation poses a particularly acute reliability challenge in large language model-based multi-agent systems (LLM-MAS), as localized errors can compound across iterative interaction rounds and compromise decision-making. Through repeated context reuse and agent communications, such minor errors can be progressively adopted and amplified, eventually leading to system-level false consensus (Laban et al., 2026; Xie et al., 2026) or causing agents to drift from their assigned roles and objectives (Shekkizhar et al., 2025). Persistent memory has been viewed as a promising mechanism for mitigating error propagation (Xiong et al., 2026; Shinn et al., 2023;

[1] https://anonymous.4open.science/r/RECOVER-8888/README.md

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.

Zhang et al., 2025c) by preserving interaction histories and task states beyond the communication context (Packer et al., 2023). Motivated by this potential and the empirical benefits of memory in single-agent LLM systems, recent studies (Zhang et al., 2025a; Wu et al., 2026; Feng et al., 2026) have incorporated explicit memory mechanisms into LLM-MAS.

Existing methods employ diverse forms of memory into LLM-MAS, including shared execution histories (Hong et al., 2024), generalizable insights and hierarchical memories of collaboration trajectories (Zhang et al., 2025a), and agent-level memory such as peer reliability for coordination (Feng et al., 2026). However, these designs lack a theoretical account of ***why persistent memory can enable recovery from propagated errors***. This missing connection leaves unclear *what information must be retained* and *when memory should intervene*. Consequently, memory systems may retain recovery-irrelevant history and invoke memory when it is unnecessary, increasing ***storage overhead*** and ***token usage*** in long-horizon MAS.

We address this gap by modeling error propagation to establish the conditions under which persistent memory enables recovery from propagated errors in LLM-MAS. Specifically, the analysis requires two steps: ***i) tracing error propagation in LLM-MAS*** and ***ii) establishing recovery limits***. These results guide ***iii) memory design*** by identifying *what information to store* and *when to inject it*, with the aim of reducing token and storage costs in long-horizon MAS.

i) Tracing error propagation in LLM-MAS. To the best of our knowledge, we develop the first theoretical characterization of ***a coupled micro–macro error propagation law*** in LLM-MAS. At *micro level* semantic errors spread through agent-to-agent (A2A) communication within each task step. (Shekkizhar et al., 2025; Laban et al., 2026). At *macro level* erroneous joint actions perturb the environment, whose feedback shapes subsequent communication and closes the micro–macro loop. The theorem bounds error accumulation across task steps in terms of communication topology, communication depth, and task horizon. Our theoretical analysis and empirical results jointly characterizes *driven factors of accumulated error propagation and how local error can escalate*.

ii) Establishing recovery limits and an exact recoverability separation. Building on the error propagation, we establish a information-theoretic lower bound for endogenous interaction, which serves as the memory-free reference process. And we establish ***an exact separation theorem*** between endogenous interaction and persistent memory. There exist tasks in which information loss induced by error propagation imposes a random-guessing error floor on endogenous interaction. However, *a minimal task-sufficient persistent memory enables zero-error recovery of the task target*.

iii) Memory for recovery. We develop ***RECOVER, a theory-guided in-task memory method for recovery*** translating preceding recoverability analysis into principled criteria for *what to retain* and *when to inject*. The mechanism is designed to recover correct decisions by supplying missing evidence when the transient context is insufficient. It maintains compact verified evidence and selectively injects only the facts needed to determine the required action preconditions. The memory is confined to the current task trial and discarded upon completion, avoiding cross-task storage accumulation. *This theory-driven design achieves superior performance without cross-task memory experience, while reducing token usage and memory storage requirements*.

In summary, this work ❶ identifies ***what*** factors govern error propagation in LLM-MAS and how severe the resulting errors can become, ❷ establishes ***why*** persistent memory is necessary when endogenous interaction cannot recover propagation-induced information loss, ❸ determines ***what*** a MAS should remember and ***when*** memory should intervene during an ongoing task, and ❹ empirically demonstrates that the resulting in-task memory can outperform even cross-task memory baselines while reducing token usage and storage overhead.

2 Related Work

Error Propagation. Recent work on interaction reliability shows that language agents can accumulate early mistakes and drift during extended exchanges. Conversation-level studies report persistent loss of information (Laban et al., 2026), A2A dialogue studies identify role and identity drift(Shekkizhar et al., 2025), and graph-based cascade analyses trace atomic errors through message dependencies(Xie et al., 2026). In LLM-MAS, error propagation is driven by two coupled processes: at the *micro level*, message passing spreads semantic errors through agent-to-agent (A2A) communication (Shekkizhar et al., 2025; Laban et al., 2026). At the *macro level*, erroneous joint actions alter subsequent environment states and observations, which feed back into later A2A interactions (Xie et al., 2026).

However, existing studies largely focus on iterative communication and A2A message passing, leaving the coupled micro–macro feedback loop insufficiently characterized. These works do not close the loop from semantic drift to joint actions, environment-state changes, and subsequent communication. To our knowledge, this paper is the first to models this coupled micro–macro error propagation process in LLM-MAS and derives its long-horizon recoverability. The proposed theorems show that, after task-relevant information is lost, further communication cannot ensure recovery, whereas a minimal persistent state can enable zero-error recovery.

Memory for LLM–MAS. Memory in LLMs can be divided into *in-task* and *cross-task* memory. Cross-task methods transfer experience beyond the current trajectory, across episodes, tasks, and sessions, as in Voyager (Wang et al., 2023), MemoryBank (Zhong et al., 2024), and GenerativeAgents (Park et al., 2023). In-task memory, by contrast, is confined to the current trajectory and retains or compresses its action–observation context without relying on cross-task experience, as in HiAgent (Hu et al., 2025), ACON (Kang et al., 2026), and the implicit interaction history maintained by ChatDev (Qian et al., 2024). Memory mechanisms for LLM-MAS remain comparatively underexplored. Existing MAS frameworks, including MetaGPT (Hong et al., 2024) and MacNet (Qian et al., 2025), primarily structure collaborative roles and communication. More recent work has begun to introduce dedicated memory mechanisms for MAS. For example, G-Memory (Zhang et al., 2025a) using hierarchical memories to organize cross-trial collaboration.

However, existing works do not determine which information is minimally sufficient to reduce system-level error after propagation-induced information loss. They also lack a principled criterion for when memory should be updated and injected. Rely on theoretical analysis, this paper connects memory content to decision identifiability, and connects memory injection to whether retained evidence resolves an undetermined action-precondition check.

3 PRELIMINARY

MAS. We consider N LLM agents solving a task over T steps. Communication follows a directed graph $\mathcal{G} = (V, E)$, whose adjacency entry $(A_{\mathcal{G}})_{i \leftarrow j} = 1$ indicates communication from agent j to agent i. At step t, s_t denotes the macro-level environment state and $c_{i,t}^{(k)}$ agent i's semantic context after round k, updated using its current context, incoming messages, and observations. After K rounds, the MAS generates a text command mapped to a canonical action $a_t \in \mathcal{A}$, whose execution yields s_{t+1} and feedback o_{t+1}. This couples within-step A2A communication (micro-loop) with action–environment interaction across steps (macro-loop).

Transient information and persistent memory. Let X_t denote the complete transient input available to the MAS, including agent contexts, accessible observations, the query, and shared task information. Let Y_t and $\widehat{Y}_t$ denote the correct and predicted task targets, and let D denote a decision time. $H(Y)$ denotes the Shannon entropy of the task target Y, and $I(Y;U)$ denotes its mutual information with the available information U, both measured in bits. Persistent memory M_t preserves earlier information from the same task and is available alongside X_t. The quantities $\mathrm{err}(Y_D \mid X_D)$ and $\mathrm{err}(Y_D \mid X_D, M_D)$ denote the minimum achievable task errors using the transient input alone and together with persistent memory, respectively.

Error in MAS. An asterisk denotes the corresponding ideal reference for the same task. We measure *micro-level semantic drift* by $\Delta_{i,t}^{(k)} := \left\| c_{i,t}^{(k)} - c_{i,t}^{(k)*} \right\|_{\mathcal{H}}$ and *macro-state systematic deviations* by $\delta_t := \|s_t - s_t^*\|$, where $\|\cdot\|_{\mathcal{H}}$ measures distance between semantic contexts. The vector $\Delta_t^{(k)} = (\Delta_{1,t}^{(k)}, \ldots, \Delta_{N,t}^{(k)})^\top$ stacks all agents' deviations and $\|\cdot\|_{\mathsf{X}}$ denotes an absolute monotone vector norm and its induced matrix norm. For the actual and ideal canonical action distributions π_t and π_t^* after communication, the action discrepancy is $e_t^a := d_{\mathcal{A}}(\pi_t, \pi_t^*)$, where $d_{\mathcal{A}}$ is a task-level distance between command distributions.

To distinguish these deviations from decision correctness, we introduce the *task error*. It measures the residual error after optimally recovering the target from the available information:

Definition 1 (Minimum achievable task error)**.** *Let $Y \in \mathcal{A}$ be the correct task target and let U denote the information available to a decision rule. The minimum achievable task error is* $\mathrm{err}(Y \mid U) := \inf_f \Pr[f(U) \neq Y]$, *where the infimum ranges over admissible decision rules f based on U.*

4 HOW DO ERRORS PROPAGATE IN MAS?

Errors in an LLM-MAS propagate through two coupled levels: the *micro loop* of A2A communication within each task step and the *macro loop* of action–environment interaction across task

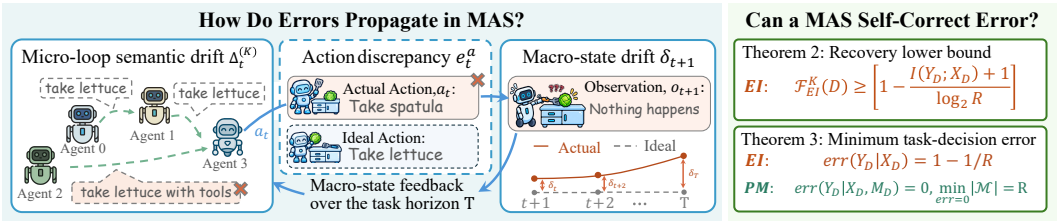

Figure 2: Errors Propagate in MAS.

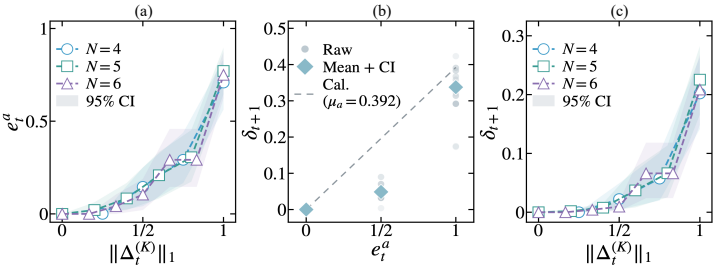

Figure 3: **Quantitative validation of the micro–macro propagation.** (a) Action discrepancy e_t^a increases with terminal semantic drift $\|\Delta_t^{(K)}\|_1$. (b) The action-to-state bound, calibrated on separate tasks and fixed during testing (Cal., $\mu_a = 0.392$), achieves high coverage with small slack on held-out tasks. (c) The resulting relationship between terminal drift and next-step macro-state deviation δ_{t+1} persists across 48 communication configurations.

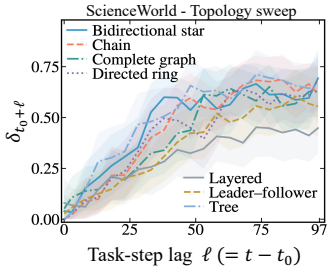

Figure 4: **Error accumulation over the task horizon T.** A single perturbation at t_0, $\delta_{t_0+\ell}$ grows across seven topologies, evidencing closed micro–macro loop error accumulation over T.

steps. The micro loop maps the initial agent-wise semantic drift $\Delta_t^{(0)}$ to the terminal semantic drift $\Delta_t^{(K)}$ over K communication rounds, during which A2A communication both transports drift across agents and introduces additional perturbations. This micro-loop semantic drift induces the action discrepancy e_t^a, which links micro-level communication error to macro-level environment error. In the macro loop, executing the resulting action affects the next environment state and its macro-state deviation δ_{t+1}. The resulting environment feedback then enters the next communication block and contributes to its initial semantic drift $\Delta_{t+1}^{(0)}$. Thereby, $\Delta_t^{(K)} \to e_t^a \to \delta_{t+1}$ closes the micro–macro propagation loop, as shown in Figure 2.

Closed-loop error propagation law. To derive the closed-loop error propagation law, we start from the semantic-drift bound for one communication round under local incremental Lipschitz conditions on message generation and context updating (Asadi et al., 2018). This bound yields a graph-dependent factor $M_\mathcal{G}$ that encodes the effect of communication topology on error propagation through within-agent drift retention and propagation along edges in the communication graph (Topping et al., 2022; Zhang et al., 2025b).

Definition 2 (Communication topology-aware operator)**.** *For uniform bounds $d, g \geq 0$ on self-retention and per-edge propagation gains, respectively, the nonnegative topology-aware operator is defined as $M_\mathcal{G} := dI + gA_\mathcal{G}$.*

Under the local sensitivity and boundedness conditions in Appendix A.1, iterating K-round semantic-drift bound and coupling it with the macro loop yields the following error propagation law over the task horizon (Ross et al., 2011).

Theorem 1 (Closed-loop MAS error propagation law)**.** *For any communication depth $K \geq 0$ and task horizon $T \geq 1$, we define $a_{\mathcal{G},K} := \mu_0 + \mu_1\left(\|M_\mathcal{G}^K\|_\mathsf{X} + \sum_{q=0}^{K-1}\|M_\mathcal{G}^q\|_\mathsf{X}\right)$, and $\gamma_{\mathcal{G},K} := \varepsilon_0 + \varepsilon_1\left(\|M_\mathcal{G}^K\|_\mathsf{X} + \sum_{q=0}^{K-1}\|M_\mathcal{G}^q\|_\mathsf{X}\right)$, where μ_0, μ_1, ε_0, and ε_1 represent Lipschitz sensitivities and bounded error residuals. Then, with probability at least $1-\zeta$, macro-state error deviation is $\delta_T \leq a_{\mathcal{G},K}^T\delta_0 + \gamma_{\mathcal{G},K}\sum_{t=0}^{T-1} a_{\mathcal{G},K}^t$.*

Proof sketch. *(i) Micro loop: Local incremental Lipschitz bounds and nonnegative matrix comparison yield the propagation factors $\|M_\mathcal{G}^K\|_\mathsf{X}$ and $\sum_{q=0}^{K-1}\|M_\mathcal{G}^q\|_\mathsf{X}$. (ii) Macro loop: Couple semantic drift to action discrepancy and macro-state deviation, then iterate the resulting affine recursion over T steps on the rollout event of probability at least $1-\zeta$. Details in Appendix A*

To examine the propagation factors in Theorem 1, we inject erroneous factual statements into natural-language messages exchanged during real task execution. This controlled intervention makes the source and timing of errors explicit, allowing their subsequent propagation across communication topologies and depths to be measured. Figure 3 supports the micro–macro coupling, and Figure 4 illustrates cross-step propagation. The theorem identifies the following factors governing error propagation, with Figure 5 providing empirical support.

Error propagation driven factors. • The communication topology factor $M_{\mathcal{G}}$ and depth K enter the one-step amplification and residual coefficients. • The task horizon T determines how these effects accumulate through the closed micro–macro loop.

$\|M_{\mathcal{G}}^K\|_{\mathsf{X}}$ propagates the drift present at the start of the communication block through all K rounds, whereas $\sum_{q=0}^{K-1} \|M_{\mathcal{G}}^q\|_{\mathsf{X}}$ accumulates perturbations injected during those rounds. Perron–Frobenius theory (Seneta, 2006) provides comparison instances attaining the spectral growth rate of $M_{\mathcal{G}}$ with communication depth K. For fixed coefficients with $\gamma_{\mathcal{G},K} > 0$, the bound converges, grows linearly, or grows exponentially in T for $a_{\mathcal{G},K} < 1$, $= 1$, or > 1, respectively. See Appendix A.6 for discussions.

From propagation to recovery. *Contraction alone does not certify recovery.* The micro–macro derivation bounds the propagation of semantic drift into macro-state deviation, while task-decision error depends on the task-relevant information retained in the transient state.

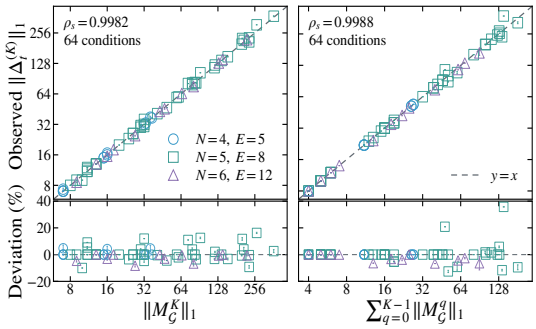

Figure 5: **Theoretical propagation factors predict the scale of semantic drift.** Across 64 communication configurations, observed terminal drift closely matches the initial-drift transport factor $\|M_{\mathcal{G}}^K\|_1$ (left) and the perturbation accumulation factor $\sum_{q=0}^{K-1} \|M_{\mathcal{G}}^q\|_1$ (right). The operator coefficients remain fixed across configurations, without configuration-specific refitting, supporting the propagation factors in Theorem 1.

5 Recovery Limits and the Necessity of Persistent Memory

This section moves from error propagation to error recovery by establishing an information-theoretic lower bound on task-decision error. We ask whether persistent memory is necessary for a MAS to recover correct decisions once propagated errors have compromised its transient state. We address this question by distinguishing *endogenous interaction* from recovery with *persistent memory*. Endogenous interaction as the memory-free reference process, which processes only the information retained in the transient state X_D. Persistent memory preserves task-relevant evidence across time for use at the final decision

Endogenous-interaction regime (EI)**.** Agents communicate and compute using only the transient state X_D, without persistent memory or additional task-relevant evidence. We formalize the information restriction (Blackwell, 1953) as follows. For each integer $K \geq 0$, let $\Pi_{\mathrm{EI}}^{(K)}$ be a nonempty class of admissible K-round protocols whose communication transcripts $Z_{D,\pi}^{1:K}$ satisfy $I\big(Y_D; Z_{D,\pi}^{1:K} \mid X_D\big) = 0, \forall\, \pi \in \Pi_{\mathrm{EI}}^{(K)}$. Under this restriction, the mutual-information chain rule (Cover & Thomas, 2005) shows endogenous interaction may process and aggregate the information retained in X_D, but cannot introduce additional information about Y_D. Consequently, every admissible protocol is subject to the same Bayes-error lower bound induced by X_D.

Persistent-memory regime (PM)**.** We augment the endogenous-interaction regime with a persistent state M_{t_0} written at some time $t_0 < D$ using only information available by t_0. The state is retained exactly until the decision time, so that $M_D = M_{t_0}$, and is available to the final decision rule in addition to the transient MAS state X_D. The MAS otherwise retains the same agents and interaction protocol. This causal-writing constraint ensures that persistent memory preserves earlier evidence rather than introducing future information.

5.1 A Lower Bound on Task Error under Endogenous Interaction

This section characterizes the recovery limit of an LLM-MAS under error propagation in the absence of persistent memory. Specifically, we derive an information-theoretic lower bound on the minimum task-decision error attainable after (K) rounds of endogenous communication and computation. The

following theorem converts the information constraint of *EI regime* into a protocol-independent lower bound on the optimal task-decision error. Since the final decision $\widehat{Y}_{D,\pi}$ is a possibly randomized post-processing of $(X_D, Z_{D,\pi}^{1:K})$ [2], the data-processing inequality (Cover & Thomas, 2005) gives $I(Y_D; \widehat{Y}_{D,\pi}) \le I(Y_D; X_D, Z_{D,\pi}^{1:K}) = I(Y_D; X_D)$. This therefore ensures that $\widehat{Y}_{D,\pi}$ cannot contain more information about Y_D than is retained in X_D. Fano's inequality converts this information ceiling into a lower bound on task-decision error (Fano & Wintringham, 1961). Because the argument holds for every admissible protocol, taking the infimum over $\Pi_{\mathrm{EI}}^{(K)}$ yields the following result.

Theorem 2 (Task error lower bound under endogenous interaction)**.** *Under the EI regime, let Y_D take values in $\mathcal{A}_R$, where $|\mathcal{A}_R| = R \ge 2$. Then, for every integer $K \ge 0$, the optimal task error $\mathcal{F}_{\mathrm{EI}}^{(K)}(D) := \inf_{\pi \in \Pi_{\mathrm{EI}}^{(K)}} \Pr\left[\widehat{Y}_{D,\pi} \ne Y_D\right]$ satisfies $\mathcal{F}_{\mathrm{EI}}^{(K)}(D) \ge \mathrm{err}(Y_D \mid X_D) \ge \left[\frac{H(Y_D) - I(Y_D;X_D) - 1}{\log_2 R}\right]_+$. **Proof**: The derivation is deferred to Appendix B.1.*

Theorem 2 shows that the error propagation induced loss of task recoverability (as in Appendix B.3) cannot be reversed through endogenous communication alone: for every $K \ge 0$, a memory-free MAS remains subject to the operational error floor $\mathrm{err}(Y_D \mid X_D)$. Additional communication may improve how the information retained in X_D is used, but it cannot reduce the error below this floor. In particular, this Bayes-error floor is strictly positive whenever $H(Y_D \mid X_D) > 0$.

5.2 Persistent Memory Enables Zero-error Recovery

Theorem 2 shows that endogenous communication cannot overcome the error floor determined by the recoverability of Y_D from X_D. The following theorem establishes an exact recovery separation between the otherwise identical *EI* and *PM* regimes on a distributed task with the error propagation induced information loss, showing that persistent memory can be necessary for zero-error recovery.

Theorem 3 (Zero-error recovery with persistent memory in a distributed task)**.** *Fix $N \ge 2$ and an action set $\mathcal{A}_R$ with $R = |\mathcal{A}_R| \ge 2$. Suppose that the MAS has aggregated all private observations by some $t_0 < D$. Then there exists a distributed task with private observations $S_1, \dots, S_N$ and target $Y_D \sim \mathrm{Unif}(\mathcal{A}_R)$ such that $X_D \perp Y_D, Y_D \perp (S_i)_{i \in J}$ for every $J \subsetneq [N]$ [3]. For this task, the following two conclusions hold.*

(i) Random-guessing error under EI. *For every $K \ge 0$, the optimal task error satisfies*

$$\mathcal{F}_{\mathrm{EI}}^{(K)}(D) = \mathrm{err}(Y_D \mid X_D) = 1 - 1/R. \tag{1}$$

(ii) Zero-error recovery with minimal PM. *There exists a persistent state $M_{t_0} = \phi(S_1, \dots, S_N)$ that is retained exactly until D so that $M_D = M_{t_0}$ and satisfies*

$$\mathrm{err}(Y_D \mid X_D, M_D) = 0, \qquad |\mathrm{supp}(M_D)| = R. \tag{2}$$

*Any finite-valued deterministic persistent state $\widetilde{M}_D$ that permits zero-error recovery satisfies $|\mathrm{supp}(\widetilde{M}_D)| \ge R$. **Proof**: See Appendix B.4.*

Theorem 3 characterizes the optimal task-decision errors of EI and PM regimes. For every communication depth K, a memory-free MAS remains at random-guessing error $1 - 1/R$. In contrast, *a causal persistent memory written before the evidence disappears achieves zero error with exactly R memory states being necessary and sufficient.* The tight R-state bound follows the classical entropy lower-bound principle for exact recovery (Karnin et al., 1983). Thus, additional communication cannot reconstruct lost task information, whereas the smallest task-sufficient persistent memory eliminates the resulting error floor.

Necessity of persistent memory for exact recovery in MAS. On the constructed MAS task, endogenous interaction remains at random-guessing error for every K, whereas a persistent state of cardinality R is necessary and sufficient for zero-error recovery.

6 What to Remember and When to Inject Memory

The preceding theorems characterize error propagation and the limits of endogenous recovery. This section now translate these results into memory design. Recovery depends on whether the system retains evidence that distinguishes the correct answer from competing alternatives. Theorem 2 bounds endogenous recovery by the information surviving in the transient state, while Theorem 3 constructs

[2] Any decoder randomness is fresh and independent of $(Y_D, X_D, Z_{D,\pi}^{1:K})$.

[3] No proper subset of agents can determine the task target.

a causal persistent memory that restores exact recovery after relevant evidence disappears. The common information-theoretic requirement is that the transient input and persistent memory jointly determine the target, leaving no residual uncertainty. The following proposition characterizes this requirement through the distinctions that memory must preserve.

Fix a decision time t and a finite set $\mathcal{H}_t$ of feasible histories under the declared state model. For each $h \in \mathcal{H}_t$, write $X_t(h)$ and $M_t(h)$ for the transient input and persistent memory along h, and $Y_t(h)$ for the correct answer to a finite-valued decision query.

Proposition 1 (Memory restores decision identifiability)**.** *A decoder g satisfying $g(X_t(h), M_t(h)) = Y_t(h)$ for every $h \in \mathcal{H}_t$ exists if and only if, for every $h, h' \in \mathcal{H}_t$,*

$$\big[X_t(h) = X_t(h') \ \wedge \ Y_t(h) \neq Y_t(h')\big] \implies M_t(h) \neq M_t(h'). \tag{3}$$

In particular, if two histories share the same transient input but require different answers, no decoder using that input alone can guarantee correctness on both.

In the construction of Theorem 3, $X_D \perp Y_D$ and the persistent state supplies precisely $I(Y_D; M_D \mid X_D) = H(Y_D) = \log_2 R$ bits. Proposition 1 expresses the same sufficiency condition for a general finite decision query. Under any distribution assigning positive probability to every history in $\mathcal{H}_t$, this condition is equivalent to $H(Y_t \mid X_t, M_t) = 0$, or, equivalently, $I(Y_t; M_t \mid X_t) = H(Y_t \mid X_t)$.

For exact recovery, Proposition 1 requires memory to distinguish histories that share the same transient input but require different answers. RECOVER operationalizes this requirement through action preconditions: for a proposed command $\tilde{a}_t$, we define the decision query as the verdict on whether all declared preconditions $\mathrm{Pre}(\tilde{a}_t)$ hold. For the command "*put lettuce 1 in/on countertop 2*", the preconditions take the form $\mathrm{Pre}(\tilde{a}_t) = \{\mathrm{held}(\mathrm{lettuce}_1), \mathrm{at}(\mathrm{agent}, \mathrm{countertop}_2)\}$. RECOVER therefor retains verified evidence that preserves these distinctions and injects it when it resolves an undetermined verdict. Proposition 3 gives sufficient conditions for exact verdict recover.

What to remember: verified evidence for action preconditions. Memory preserves the historical evidence needed to distinguish different precondition verdicts when the transient input alone is insufficient. Such evidence is retained as compact facts verified by environment observations before its support disappears from the transient context, and can later be reused for subsequent precondition checks. At each step t, the persistent memory M is updated with valid evidence verified by the resulting environment observation o_{t+1} after executing a_t:

$$M_{t+1} = \mathrm{Update}(M_t, \mathrm{Extract}_{\mathrm{Pre}}(a_t, o_{t+1}))\,. \tag{4}$$

Here, $\mathrm{Extract}_{\mathrm{Pre}}$ records newly confirmed state facts and their supporting observations in ΔE_{t+1}, including records such as $\mathrm{held}(\mathrm{lettuce}_1) = \mathrm{false}$, o_{t+1} when o_{t+1} confirms a successful placement action. Update replaces outdated state of facts in M_t which are invalidated by the latest transition. For example, replace $\mathrm{at}(\mathrm{agent}) = \mathrm{countertop}_1$ with $\mathrm{at}(\mathrm{agent}) = \mathrm{fridge}_1$ after new valid evidence come in. Thus, persistent memory retains valid facts after their evidence is loss in the transient input, following the causal construction in Theorem 3.

When to inject: resolving undetermined precondition checks. Memory is injected only when a precondition verdict is undetermined from the current evidence, but becomes determined after incorporating relevant stored evidence. Theorem 2 limits recovery by the information remaining in the transient input, while Proposition 1 provides sufficient conditions for determining the auxiliary verdict. The corresponding memory injection can be determined as:

$$\mathrm{Insert}_t = \mathbf{1}[\mathrm{Check}(\mathrm{Pre}(\tilde{a}_t), E_t) = \perp \ \wedge \ \mathrm{Check}(\mathrm{Pre}(\tilde{a}_t), E_t \cup M_t) \neq \perp]\,. \tag{5}$$

Check returns true when all declared preconditions are supported, false when at least one is refuted, and $\perp$ when the answer remains unknown. Only the relevant facts $M_t^{inj} \subseteq M_t \setminus E_t$, rather than the entire memory are selected for the current injection. For example, suppose the current evidence contains $\mathrm{held}(\mathrm{lettuce}_1) = \mathrm{true}$ but leaves $\mathrm{at}(\mathrm{agent}, \mathrm{countertop}_1) = unknown$, while $M_t^{\mathrm{inj}} : \mathrm{at}(\mathrm{agent}, \mathrm{countertop}_1) = \mathrm{true} \in M_t$. Then the precondition check changes from $\mathrm{Check}(\mathrm{Pre}(\tilde{a}_t), E_t) = \perp$ to $\mathrm{Check}(\mathrm{Pre}(\tilde{a}_t), E_t \cup M_t^{\mathrm{inj}}) = \mathrm{true}$ after injecting this relevant fact. Details are provided in Appendix C, showing the practical implementation. Experiments and injection-timing ablations show that this theory-guided memory design yields better task performance with lower token and storage costs.

7 Experiment

In this section, we evaluate three questions: **(RQ1)** How does RECOVER, theory-guided memory, compare with existing memory baselines? **(RQ2)** How does memory injection timing affect performance and token usage? **(RQ3)** How efficient is RECOVER in token usage and memory storage?

Table 1: Performance Comparison with single and multi-agent memory baselines across in-task and cross-task settings on Qwen3-14b (mean ± standard deviation).

MAS	Memory	ALFWorld	FEVER	SciWorld	Office	HotpotQA
AutoGen	Raw history	75.597±1.69	59.545±1.04	63.836±5.18	42.105±1.07	33.643±4.23
	GMemory	89.552±2.77	64.817±5.65	61.390±5.31	43.820±1.39	38.674±2.66
	MetaGPT	88.060±1.69	66.659±2.24	65.678±3.61	43.853±4.27	38.491±2.21
	Voyager	83.582±1.55	63.319±3.46	64.003±5.34	48.864±1.10	37.740±3.53
	Generative	86.567±2.09	60.201±3.96	65.987±1.75	46.067±2.96	36.531±2.62
	MemoryBank	75.373±3.18	58.087±1.03	59.125±4.00	41.558±3.76	33.190±2.65
	ChatDev	79.851±1.17	57.508±1.21	61.802±4.21	45.652±3.19	34.655±4.79
	ACON	84.328±2.10	62.747±2.49	67.934±2.71	38.298±3.20	31.071±2.01
	HiAgent	68.657±2.86	41.667±1.42	65.708±0.18	43.421±1.86	32.365±1.59
	RECOVER	**91.542**±1.46	**68.331**±0.81	**72.899**±1.83	**50.575**±1.39	**40.969**±3.62
DyLAN	Raw history	78.358±1.57	48.481±1.47	45.272±1.98	42.105±5.90	32.711±2.09
	GMemory	87.313±0.23	59.012±2.70	52.296±1.00	47.727±1.60	34.734±4.85
	MetaGPT	83.582±3.59	59.411±3.58	47.290±2.88	46.739±0.52	40.546±3.41
	Voyager	79.851±2.38	60.905±3.47	46.855±2.65	43.820±1.22	38.070±2.19
	Generative	79.851±1.79	57.201±3.36	50.846±3.09	47.368±1.36	34.477±5.23
	MemoryBank	76.866±2.15	57.048±0.95	31.116±3.72	42.857±1.69	36.278±5.43
	ChatDev	79.104±2.65	61.944±2.12	51.452±3.95	44.086±1.33	37.248±3.07
	ACON	74.627±1.37	62.466±4.83	53.194±5.24	41.489±2.63	33.004±3.63
	HiAgent	69.403±1.83	50.585±1.09	42.839±1.25	41.791±5.57	33.939±2.34
	RECOVER	**92.537**±1.74	**64.489**±3.43	**56.805**±3.56	**49.330**±3.35	**41.907**±2.36
MacNet	Raw history	61.642±2.45	59.769±1.54	59.718±1.02	36.545±1.35	38.174±2.18
	GMemory	73.091±1.32	54.838±0.57	68.192±2.71	41.331±1.52	39.257±1.55
	MetaGPT	51.452±0.82	51.799±0.17	65.816±1.06	38.273±2.34	33.024±4.75
	Voyager	67.209±2.94	60.390±1.30	56.293±1.13	39.508±3.33	38.230±3.55
	Generative	70.056±1.40	56.293±4.09	67.414±0.74	40.854±4.48	39.726±3.60
	MemoryBank	67.559±1.96	46.329±5.14	64.696±4.63	37.830±2.43	37.315±2.42
	ChatDev	60.584±1.41	54.355±3.64	63.255±2.95	38.861±2.83	36.284±3.19
	ACON	66.111±1.55	44.201±4.60	66.022±0.35	32.493±0.53	35.451±2.33
	HiAgent	63.105±1.54	50.089±1.85	59.223±1.25	33.319±5.52	34.586±3.20
	RECOVER	**77.612**±2.33	**62.462**±3.90	**71.890**±2.23	**43.834**±3.95	**41.669**±4.16

7.1 Experiment Setup

Datasets and Benchmarks. We evaluate our method on five benchmarks across four domains: embodied interaction (ALFWorld (Shridhar et al., 2021) and ScienceWorld (Wang et al., 2022)), knowledge-intensive reasoning (FEVER (Thorne et al., 2018) and HotpotQA (Yang et al., 2018)), and office automation (OfficeBench (Wang et al., 2024)).

Baselines. We organize the comparisons along two dimensions, agent scope and memory scope. ***In-task memory*** is restricted to the current task trajectory and discarded after task completion, whereas ***cross-task memory*** retains or reuses experience across task instances. The no-memory setting serves as the memory-free reference. For ***single-agent memory***, we compare in-task methods, including HiAgent and ACON (Hu et al., 2025; Kang et al., 2026), with cross-task methods, including Voyager, MemoryBank, and Generative Agents (Wang et al., 2023; Zhong et al., 2024; Park et al., 2023). For ***multi-agent memory***, we evaluate G-Memory (Zhang et al., 2025a). Following its experimental setup, we also include memory-augmented variants of MetaGPT, ChatDev, and MacNet (Hong et al., 2024; Qian et al., 2024; 2025).

Implementations Following G-Memory (Zhang et al., 2025a), we integrate memory methods with three representative multi-agent frameworks: AutoGen (Wu et al., 2024), DyLaN (Liu et al., 2024), and MacNet (Qian et al., 2025). Unless otherwise specified, we run each experiment twice using Qwen3-14B as the LLM backbone, served locally with vLLM. Qwen3-8B and Qwen3-32B are used for supplementary scale-transfer experiments. Details are provided in Appendix E.1.

7.2 Experiment Results

(RQ1) What a MAS remembers matters. Table 1 reports the highest mean performance for our method in all 15 combinations of five benchmarks and three MAS frameworks, with relative improvements of 1.45%–7.31% over the strongest memory baseline in both in-task and cross-task settings. Relative to raw history, the largest improvement is on FEVER with DyLAN, where performance achieves a 33.02% relative gain. **RECOVER** ***achieves the best task performance among without relying on cross-task experience.***

Across all datasets and LLM backbones in Table 1 and Figure 7, we observe cases where memory underperforms raw history. On ALFWorld with AutoGen, for instance, raw history achieves 75.597%, compared with 68.657% for HiAgent, while our precondition-evidence memory reaches 91.542%. This pattern shows that memory effectiveness depends critically on what is retained. These comparisons evaluate each memory system as a complete procedure. ***The value of mem-***

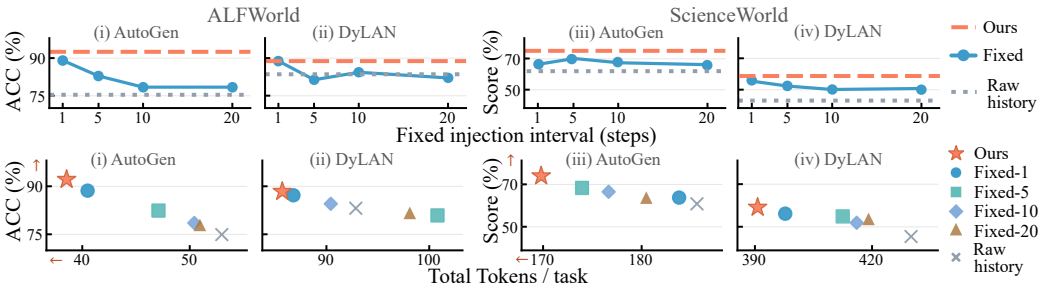

Figure 6: Effect of memory injection timing on ALFWorld and ScienceWorld with AutoGen and DyLAN. **Top:** Accuracy under fixed injection intervals. Dashed and dotted lines denote our adaptive schedule and raw history. **Bottom:** Accuracy versus total tokens per task, with stars denoting our method. **Memory injection timing of RECOVER** achieves higher accuracy than the best fixed schedule while using the fewest tokens across all settings.

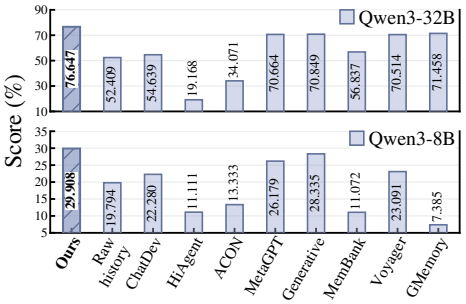

Figure 7: ScienceWorld performance with DyLAN using Qwen3-32B (top) and Qwen3-8B (bottom). **RECOVER** outperforms all baselines across different LLMs backbones.

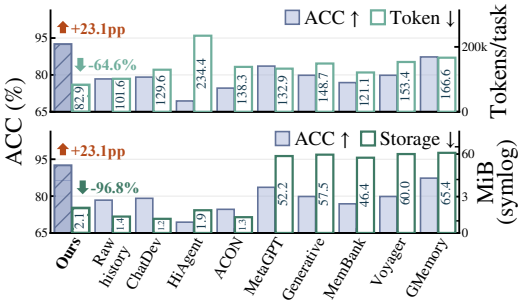

Figure 8: Accuracy and memory efficiency on ALFWorld with DyLAN. Hollow bars show per-task token usage (top) and peak memory size (bottom). **RECOVER** achieves the highest accuracy with lower token and storage costs.

ory lies in preserving evidence that keeps decisions recoverable when the transient context is no longer sufficient.

(RQ2) Memory injection timing matters. Our method matches or exceeds the accuracy of all fixed injection schedules while using the fewest total tokens across all four settings. On ScienceWorld, it improves accuracy by 5.56 pp over every-step injection and reduces total token usage by 7.6%. The token reduction comes from fewer memory insertions and shorter trajectories, which reduce repeated calls and accumulated history. ***Theory-guided injection timing improves task performance while reducing end-to-end token use.***

(RQ3) Memory Efficiency Analysis. In the ALFWorld efficiency comparison in Figure 8, our method achieves the highest task success rate and the lowest total token use under DyLAN. **RECOVER** uses 82.9k tokens per task, compared with 234.4k for HiAgent, reducing token usage by approximately 64.6%, while improving task performance by 23.134 percentage points The peak serialized memory payload is 2.06 MiB under DyLAN, compared with 65.4 MiB for G-Memory, a reduction of approximately 97%. These results support our design of retaining compact evidence within the current task and injecting it selectively. ***Theory-driven retained memory reduces token and storage overhead while improving task performance.***

8 Conclusion

We connect error propagation, recovery limits, and persistent memory in LLM-MAS. Our closed-loop analysis characterizes how communication topology, communication depth, and task horizon affect error accumulation. A constructed distributed task shows that causal persistent memory enables zero-error recovery while endogenous interaction remains at random-guessing error. These results guide RECOVER in what to retain and when to inject memory. Experiments across five benchmarks and three MAS frameworks show improved performance over in-task and cross-task memory baselines, with lower token and storage costs. These findings support decision recoverability as a guiding principle for efficient memory design in LLM-MAS. Limitations: Future work could extend the recovery analysis to imperfect memory retention and noisy evidence verification.

AI ASSISTANCE

Generative large language models were used responsibly in this study to support related works retrieval and discovery, manuscript editing and code development. All AI-assisted outputs were thoroughly reviewed and, when appropriate, revised by the authors to ensure their accuracy, reliability, and consistency with the authors' intended meaning, underlying data, and methodological framework.

REPRODUCIBILITY STATEMENT

We have made every effort to ensure that the results presented in this paper are reproducible. All code and datasets are made publicly available in an anonymous repository to facilitate replication and verification.

REFERENCES

Kavosh Asadi, Dipendra Misra, and Michael Littman. Lipschitz continuity in model-based reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 264–273, 2018.

David Blackwell. Equivalent comparisons of experiments. *The Annals of Mathematical Statistics*, 24(2):265–272, June 1953.

Thomas M. Cover and Joy A. Thomas. *Entropy, Relative Entropy, and Mutual Information*, chapter 2,11, pp. 13–55, 347–408. 2005.

Robert M Fano and WT Wintringham. Transmission of information, 1961.

Peilin Feng, Suorong Yang, and Soujanya Poria. Σ-Mem: An online reliability memory for LLM-based multi-agent systems. *arXiv preprint*, 2026.

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.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. HiAgent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32779–32798, July 2025.

Minki Kang, Wei-Ning Chen, Dongge Han, Huseyin A Inan, Lukas Wutschitz, Yanzhi Chen, Robert Sim, and Saravan Rajmohan. ACON: Optimizing context compression for long-horizon LLM agents. In *Forty-third International Conference on Machine Learning*, 2026.

Ehud D. Karnin, Jonathan W. Greene, and Martin E. Hellman. On secret sharing systems. *IEEE Transactions on Information Theory*, 29(1):35–41, January 1983.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. Llms get lost in multi-turn conversation. In *International Conference on Learning Representations*, volume 2026, pp. 54738–54778, 2026.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic LLM-powered agent network for task-oriented agent collaboration. In *First Conference on Language Modeling*, 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*, UIST '23, 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.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011.

E. Seneta. *Non-negative Matrices and Markov Chains*. Springer Series in Statistics. 2nd edition, 2006.

Sarath Shekkizhar, Romain Cosentino, Adam Earle, and Silvio Savarese. Echoing: Identity failures when llm agents talk to each other. *arXiv preprint*, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 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.

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.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding Over-Squashing and Bottlenecks on Graphs via Curvature. In *International Conference on Learning Representations*, 2022.

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*, 2023.

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, December 2022.

Zilong Wang, Yuedong Cui, Li Zhong, Zimin Zhang, Da Yin, Bill Yuchen Lin, and Jingbo Shang. Officebench: Benchmarking language agents across multiple applications for office automation. *arXiv preprint arXiv:2407.19056*, 2024.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen LLM applications via multi-agent conversation. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024.

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*, 2026.

Yizhe Xie, Congcong Zhu, Xinyue Zhang, Tianqing Zhu, Dayong Ye, Minfeng Qi, Huajie Chen, and Wanlei Zhou. From spark to fire: Modeling and mitigating error cascades in llm-based multi-agent collaboration. *arXiv preprint*, 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. pp. 623–645, July 2026.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Guibin Zhang, Muxin Fu, Kun Wang, Guancheng Wan, Miao Yu, and Shuicheng YAN. G-memory: Tracing hierarchical memory for multi-agent systems. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. G-Designer: Architecting Multi-agent Communication Topologies via Graph Neural Networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76678–76692. PMLR, 2025b.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model-based agents. *ACM Trans. Inf. Syst.*, 43(6), 2025c.

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.

APPENDIX CONTENTS

A. Error Propagation (Proofs for Section 4)

A.1 Standing Conditions used in the proof 14
A.2 One-Round Micro Semantic Drift Recursion 15
A.3 Propagation over K Communication Rounds 16
A.4 Closing the Micro-Macro Feedback Loop 17
A.5 Proof of the Closed-loop MAS Error Propagation Law 17
A.6 Propagation Regimes and Spectral Attainability 18

B. Recovery Limits and Persistent Memory Sufficiency (Proofs for Section 5)

B.1 Endogenous Interaction and the Information Constraint 19
B.2 Distributed Task and Causal Persistent State 20
B.3 An Error Propagation Induced Information Loss 21
B.4 Proof of the Theorem 3 22
B.5 Proof of Memory Decision Identifiability 22
B.6 Sufficient Conditions for Precondition Recovery 23

C. RECOVER: Procedure and Benchmark Instantiations

C.1 Memory Procedure 24
C.2 Case Studies and Benchmark Instantiations 24

D. Experimental validation of Proposed Theorems

D.1 Experimental setup and calibration 28
D.2 Figure 3:Local Micro-Macro Coupling 29
D.3 Figure 4:Finite-Horizon Persistence of Perturbations 32
D.4 Figure 5:Topology-Depth Propagation Factors 33
D.5 Connection to the Closed-Loop Bound 34
D.6 Shared Implementation and Statistical Protocol 35

E. Experimental Setup and Evaluation Protocols

E.1 Experimental Setup: Benchmarks and Implementation 36
E.2 Additional Experimental Results 38

A Error Propagation (Proofs for Section 4)

This appendix gives the proof of the close-coupled error propagation law in Section 4. The proof follows the usual local Lipschitz analysis of sequential decision systems and transition models (Asadi et al., 2018; Ross et al., 2011). It is organized around the propagation pipeline in the main text. We first state the rollout conditions, then derive the one-round semantic-drift recursion, unroll it over K communication rounds, close the action–state feedback loop, and finally prove the task-horizon bound and its attainable regimes.

A.1 Standing conditions

Representations and reference rollouts. Fix task-specific encodings of raw contexts and messages into normed spaces, and a state representation $s_t \in \mathcal{S}$. Write $c_{i,t}^{(k)} = \phi_{\mathcal{H}}(\text{context}_{i,t}^{(k)}) \in \mathcal{H}$; messages below likewise denote their encoded values. All sensitivities and residual bounds refer to these fixed representations. Starred quantities follow the specified reference process for the same task. Fix a joint law for the actual and reference rollouts, including the coupling of their exogenous randomness; unmatched effects are covered by the residual bounds. Initial deviations are allowed through δ_0. Appendix D specifies the experimental representations and references; the local terminal-proposal representation uses $c_i = b_i/N$, where b_i marks a replaced proposal and $m = \sum_i b_i$, giving $\|\Delta_t^{(K)}\|_1 = m/N$.

Action and state distances. A fixed canonicalizer $\kappa : \text{Text} \to \mathcal{A}$ induces the canonical action laws π_t and π_t^*. For a command cost $\ell_{\mathcal{A}}$, a compatible choice of $d_{\mathcal{A}}$ is the infimum of expected command cost over all couplings of these laws. For point masses, this gives $d_{\mathcal{A}}(\delta_a, \delta_{a^*}) = \ell_{\mathcal{A}}(a, a^*)$, so deterministic decoding identifies distributional discrepancy with the paired-command cost. Under stochastic decoding, sampled command costs instead depend on the chosen coupling and require a corresponding estimator to measure the law-level discrepancy. The proof also applies with $\delta_t = d_{\mathcal{S}}(s_t, s_t^*)$ for a fixed state distance, provided every state-norm term in the sensitivity conditions is replaced by that distance: the derivation uses states only through these conditions. This covers the local experiment's Jaccard and goal-progress state distance under its corresponding interface conditions.

Rollout event. Fix $T \geq 1$ and $K \geq 0$. Let $\mathcal{E}_{T,K}$ be the event on which the following conditions hold throughout the task-reachable region of the paired rollouts, at every step $0 \leq t < T$ and, for communication conditions, every round $0 \leq k < K$. Assume $\Pr(\mathcal{E}_{T,K}) \geq 1 - \zeta$ under the fixed joint law. The propagation inequalities hold pathwise on this event; empirical coverage reported for sampled comparisons is a separate evaluation quantity.

Message generation and context update. For an edge $j \to i$, write the actual and ideal messages as

$$m_{j\to i,t}^{(k)} = \Psi_{j\to i}^{(k)}(c_{j,t}^{(k)}, s_t) + \omega_{j\to i,t}^{(k)}, \qquad m_{j\to i,t}^{(k)*} = \Psi_{j\to i}^{(k)}(c_{j,t}^{(k)*}, s_t^*).$$

On the task-reachable region, assume

$$\|\Psi_{j\to i}^{(k)}(c, s) - \Psi_{j\to i}^{(k)}(c', s')\| \leq M_{ij}^{c,k}\|c - c'\| + M_{ij}^{s,k}\|s - s'\|.$$

The receiver update is

$$c_{i,t}^{(k+1)} = \Phi_i^{(k)}\Big(c_{i,t}^{(k)}, \{m_{j\to i,t}^{(k)}\}_{j\in\mathcal{N}_{\text{in}}(i)}, s_t\Big) + \rho_{i,t}^{(k)},$$

and the ideal update uses starred quantities. We assume the local bound

$$\begin{aligned}&\Big\|\Phi_i^{(k)}(c_i, \{m_{j\to i}\}, s) - \Phi_i^{(k)}(c_i', \{m_{j\to i}'\}, s')\Big\| \\ &\quad\leq L_i^{c,k}\|c_i - c_i'\| + \sum_{j\in\mathcal{N}_{\text{in}}(i)} L_{ij}^{m,k}\|m_{j\to i} - m_{j\to i}'\| + L_i^{s,k}\|s - s'\|.\end{aligned}$$

These are local incremental sensitivity conditions for the message and context maps, as used in Lipschitz analyses of sequential systems (Asadi et al., 2018).

Graph envelope and bounded inputs. Let $W_t^{(k)}$, $B_t^{(k)}$, and $\varepsilon_t^{(k)}$ denote the nonnegative coefficient matrix, macro-state input, and residual vector induced by the preceding local bounds. Assume that, for some $d, g, b, \bar{\varepsilon} \geq 0$,

$$0 \preceq W_t^{(k)} \preceq M_{\mathcal{G}} := dI + gA_{\mathcal{G}}, \qquad \|B_t^{(k)}\|_{\mathsf{X}} \leq b, \qquad \|\varepsilon_t^{(k)}\|_{\mathsf{X}} \leq \bar{\varepsilon}.$$

Here d bounds self-retention and g bounds propagation along a communication edge. This graph-level envelope is the standard comparison step used to turn local edge sensitivities into a topology-dependent bound (Topping et al., 2022; Zhang et al., 2025b).

Initialization and action–state interfaces. Assume, on $\mathcal{E}_{T,K}$,

$$\|\Delta_t^{(0)}\|_{\mathsf{X}} \leq c_0\delta_t + \bar{\xi}, \qquad c_0, \bar{\xi} \geq 0,$$

and

$$e_t^a \leq \lambda_s\delta_t + \lambda_c\|\Delta_t^{(K)}\|_{\mathsf{X}} + \bar{\nu}, \qquad \delta_{t+1} \leq \mu_s\delta_t + \mu_a e_t^a + \bar{\eta},$$

where all displayed coefficients are nonnegative. The first inequality closes the macro-to-micro part of the pipeline; the latter two inequalities are the canonical-action and macro-transition sensitivity conditions.

Throughout, $\|\cdot\|_{\mathsf{X}}$ is an absolute monotone vector norm and its induced matrix norm. Thus $0 \leq x \preceq y$ implies $\|x\|_{\mathsf{X}} \leq \|y\|_{\mathsf{X}}$, and $\|Bx\|_{\mathsf{X}} \leq \|B\|_{\mathsf{X}}\|x\|_{\mathsf{X}}$.

A.2 One-round micro semantic-drift recursion

This subsection proves the local recursion used by the communication block. The proof applies the triangle inequality to message generation and then the local sensitivity of the receiver update. It makes explicit which terms transport existing semantic drift, which term comes from the macro state, and which terms are residual perturbations.

Lemma 1 (One-round semantic-drift recursion)**.** *Under the standing conditions in Appendix A.1, for every t and k on $\mathcal{E}_{T,K}$,*

$$\Delta_t^{(k+1)} \preceq W_t^{(k)}\Delta_t^{(k)} + B_t^{(k)}\delta_t + \varepsilon_t^{(k)}.$$

The entries of the three nonnegative terms can be chosen as

$$(W_t^{(k)})_{i\leftarrow i} = L_i^{c,k}, \qquad (W_t^{(k)})_{i\leftarrow j} = L_{ij}^{m,k}M_{ij}^{c,k},$$
$$(B_t^{(k)})_i = L_i^{s,k} + \sum_{j\in\mathcal{N}_{\mathrm{in}}(i)} L_{ij}^{m,k}M_{ij}^{s,k}, \qquad (\varepsilon_t^{(k)})_i = \|\rho_{i,t}^{(k)}\| + \sum_{j\in\mathcal{N}_{\mathrm{in}}(i)} L_{ij}^{m,k}\|\omega_{j\to i,t}^{(k)}\|,$$

where the off-diagonal entry is zero when $j \notin \mathcal{N}_{\mathrm{in}}(i)$.

The lemma is a direct consequence of the local incremental Lipschitz bounds; no horizon-level argument is used here.

Proof. Fix an incoming edge $j \to i$. The message bound and the triangle inequality give

$$\|m_{j\to i,t}^{(k)} - m_{j\to i,t}^{(k)*}\| \leq M_{ij}^{c,k}\Delta_{j,t}^{(k)} + M_{ij}^{s,k}\delta_t + \|\omega_{j\to i,t}^{(k)}\|.$$

Applying the receiver-update sensitivity bound yields

$$\Delta_{i,t}^{(k+1)} \leq L_i^{c,k}\Delta_{i,t}^{(k)} + \sum_{j\in\mathcal{N}_{\mathrm{in}}(i)} L_{ij}^{m,k}\|m_{j\to i,t}^{(k)} - m_{j\to i,t}^{(k)*}\| + L_i^{s,k}\delta_t + \|\rho_{i,t}^{(k)}\|.$$

Substituting the message bound and collecting the coefficients of $\Delta_{j,t}^{(k)}$, δ_t, and the residuals gives the stated componentwise recursion. Stacking the N component inequalities proves the lemma. □

A.3 PROPAGATION OVER K COMMUNICATION ROUNDS

The previous lemma describes one micro-loop update. This subsection unrolls it over the K rounds in one task step. Definition 2 introduces the communication topology-aware operator $M_{\mathcal{G}} = dI + gA_{\mathcal{G}}$ in the main text. Since $W_t^{(k)} \preceq M_{\mathcal{G}}$, this operator bounds the propagation of existing semantic drift in the one-round recursion of Lemma 1. This subsection proves the K-round semantic-drift bound by composing the same operator over one task step, yielding the two matrix-norm factors in Theorem 1.

Definition 3 (Communication propagation factors)**.** *For $M_{\mathcal{G}}$ from Definition 2 and a communication depth $K \geq 0$, define the terminal propagation factor and the accumulated injection factor by*

$$Q_{\mathcal{G}}(K) := \|M_{\mathcal{G}}^K\|_{\mathsf{X}}, \qquad S_{\mathcal{G}}(K) := \sum_{q=0}^{K-1} \|M_{\mathcal{G}}^q\|_{\mathsf{X}},$$

with $S_{\mathcal{G}}(0) = 0$.

The notation $Q_{\mathcal{G}}(K)$ and $S_{\mathcal{G}}(K)$ abbreviates the two matrix-norm terms in the main theorem. The first bounds propagation of initial drift through all K rounds. The second bounds the accumulated effect of perturbations injected during communication, each of which propagates through only the remaining rounds.

To explain how the topology in Definition 2 enters these factors, we expand the powers of $M_{\mathcal{G}}$. Because I commutes with $A_{\mathcal{G}}$, the binomial theorem gives

$$M_{\mathcal{G}}^q = \sum_{r=0}^{q} \binom{q}{r} d^{q-r} g^r A_{\mathcal{G}}^r.$$

Under the convention $(A_{\mathcal{G}})_{i \leftarrow j} = 1$ when j sends to i, the entry $(A_{\mathcal{G}}^r)_{i \leftarrow j}$ counts directed communication walks of length r from j to i. The expansion weights these walks by the self-retention and per-edge propagation gains d and g from the main-text definition. Taking the induced norms gives the propagation bounds summarized by $Q_{\mathcal{G}}(K)$ and $S_{\mathcal{G}}(K)$.

Lemma 2 (K-round semantic-drift bound)**.** *Under the standing conditions and on $\mathcal{E}_{T,K}$, for every $0 \leq t < T$,*

$$\|\Delta_t^{(K)}\|_{\mathsf{X}} \leq Q_{\mathcal{G}}(K)(c_0\delta_t + \bar{\xi}) + S_{\mathcal{G}}(K)(b\delta_t + \bar{\varepsilon}).$$

The proof is an unrolling of Lemma 1, followed by the monotonicity and submultiplicativity of the induced norm.

Proof. Repeatedly applying Lemma 1 gives

$$\begin{aligned}\Delta_t^{(K)} \preceq\ & \left(W_t^{(K-1)} \cdots W_t^{(0)}\right) \Delta_t^{(0)} \\ & + \sum_{\ell=0}^{K-1} \left(W_t^{(K-1)} \cdots W_t^{(\ell+1)}\right) \left(B_t^{(\ell)}\delta_t + \varepsilon_t^{(\ell)}\right),\end{aligned}$$

where an empty product is the identity. Since all matrices are nonnegative and $W_t^{(k)} \preceq M_{\mathcal{G}}$,

$$W_t^{(K-1)} \cdots W_t^{(0)} \preceq M_{\mathcal{G}}^K, \qquad W_t^{(K-1)} \cdots W_t^{(\ell+1)} \preceq M_{\mathcal{G}}^{K-1-\ell}.$$

Therefore,

$$\begin{aligned}\|\Delta_t^{(K)}\|_{\mathsf{X}} \leq\ & Q_{\mathcal{G}}(K)\|\Delta_t^{(0)}\|_{\mathsf{X}} \\ & + \sum_{\ell=0}^{K-1} \|M_{\mathcal{G}}^{K-1-\ell}\|_{\mathsf{X}} \left(\|B_t^{(\ell)}\|_{\mathsf{X}}\delta_t + \|\varepsilon_t^{(\ell)}\|_{\mathsf{X}}\right).\end{aligned}$$

Using the bounds in Appendix A.1, changing variables to $q = K - 1 - \ell$, and using the definitions of $Q_{\mathcal{G}}(K)$ and $S_{\mathcal{G}}(K)$ proves the lemma. □

A.4 Closing the micro–macro feedback loop

This subsection proves how the communication topology factors from Appendix A.3 enter the next macro-state deviation. We compose the local action and macro-transition sensitivity bounds in Appendix A.1, following the Lipschitz transition analysis of Asadi et al. (2018). Applying Lemma 2 gives a one-step bound that retains the separate contributions of $Q_{\mathcal{G}}(K)$ and $S_{\mathcal{G}}(K)$. Collecting its coefficients then gives the affine recursion used in Theorem 1.

Lemma 3 (Factor-resolved one-step recursion)**.** *Under the conditions in Appendix A.1, on $\mathcal{E}_{T,K}$ and for every $0 \le t < T$,*

$$\begin{aligned}\delta_{t+1} \le \Big[\mu_0 + \mu_a\lambda_c\big(c_0 Q_{\mathcal{G}}(K) + bS_{\mathcal{G}}(K)\big)\Big]\delta_t \\ + \varepsilon_0 + \mu_a\lambda_c\big(\bar{\xi}Q_{\mathcal{G}}(K) + \bar{\varepsilon}S_{\mathcal{G}}(K)\big),\end{aligned}$$

where

$$\mu_0 := \mu_s + \mu_a\lambda_s, \qquad \varepsilon_0 := \mu_a\bar{\nu} + \bar{\eta}.$$

The proof first maps terminal semantic drift to action discrepancy and then maps action discrepancy to the next macro-state deviation. The two topology factors remain visible throughout this calculation.

Proof. Step 1. From semantic drift to action discrepancy. Fix $0 \le t < T$ and work on $\mathcal{E}_{T,K}$. Substituting Lemma 2 into the action-sensitivity bound gives

$$\begin{aligned}e_t^a &\le \lambda_s\delta_t + \lambda_c\|\Delta_t^{(K)}\|_{\mathsf{X}} + \bar{\nu} \\ &\le \lambda_s\delta_t + \lambda_c Q_{\mathcal{G}}(K)(c_0\delta_t + \bar{\xi}) \\ &\quad + \lambda_c S_{\mathcal{G}}(K)(b\delta_t + \bar{\varepsilon}) + \bar{\nu}.\end{aligned}$$

Thus the action discrepancy depends on both the initial drift propagated through all K rounds and the perturbations accumulated during communication.

Step 2. From action discrepancy to macro-state deviation. The macro-transition bound maps this action discrepancy to the next state. Substituting the preceding inequality gives

$$\begin{aligned}\delta_{t+1} &\le \mu_s\delta_t + \mu_a e_t^a + \bar{\eta} \\ &\le \big[\mu_s + \mu_a\lambda_s + \mu_a\lambda_c\{c_0 Q_{\mathcal{G}}(K) + bS_{\mathcal{G}}(K)\}\big]\delta_t \\ &\quad + \mu_a\bar{\nu} + \bar{\eta} + \mu_a\lambda_c\{\bar{\xi}Q_{\mathcal{G}}(K) + \bar{\varepsilon}S_{\mathcal{G}}(K)\}.\end{aligned}$$

Substituting $\mu_0 = \mu_s + \mu_a\lambda_s$ and $\varepsilon_0 = \mu_a\bar{\nu} + \bar{\eta}$ proves the lemma. □

A.5 Proof of the closed-loop MAS error propagation law

The preceding one-step bound and the coefficient bounds that follow it supply one affine error map for one task step. This subsection proves the main theorem by composing the same map over the task horizon. The argument is the standard iteration of a scalar affine recursion.

Proof of Theorem 1. On $\mathcal{E}_{T,K}$, Lemma 3 gives

$$\begin{aligned}\delta_{t+1} \le \Big[\mu_0 + \mu_a\lambda_c\big(c_0 Q_{\mathcal{G}}(K) + bS_{\mathcal{G}}(K)\big)\Big]\delta_t \\ + \varepsilon_0 + \mu_a\lambda_c\big(\bar{\xi}Q_{\mathcal{G}}(K) + \bar{\varepsilon}S_{\mathcal{G}}(K)\big).\end{aligned}$$

Because $Q_{\mathcal{G}}(K)$, $S_{\mathcal{G}}(K)$, and δ_t are nonnegative,

$$\begin{aligned}\delta_{t+1} \le \Big[\mu_0 + (c_0 + b)\mu_a\lambda_c\big(Q_{\mathcal{G}}(K) + S_{\mathcal{G}}(K)\big)\Big]\delta_t \\ + \varepsilon_0 + (\bar{\xi} + \bar{\varepsilon})\mu_a\lambda_c\big(Q_{\mathcal{G}}(K) + S_{\mathcal{G}}(K)\big).\end{aligned}$$

Using

$$\mu_1 := (c_0 + b)\mu_a\lambda_c, \qquad \varepsilon_1 := (\bar{\xi} + \bar{\varepsilon})\mu_a\lambda_c,$$

and the definitions

$$a_{\mathcal{G},K} := \mu_0 + \mu_1\big(Q_{\mathcal{G}}(K) + S_{\mathcal{G}}(K)\big),$$

$$\gamma_{\mathcal{G},K} := \varepsilon_0 + \varepsilon_1\big(Q_{\mathcal{G}}(K) + S_{\mathcal{G}}(K)\big),$$

we obtain

$$\delta_{t+1} \le a_{\mathcal{G},K}\delta_t + \gamma_{\mathcal{G},K}, \qquad 0 \le t < T.$$

Iterating this affine recursion yields

$$\delta_T \le a_{\mathcal{G},K}^T \delta_0 + \gamma_{\mathcal{G},K} \sum_{t=0}^{T-1} a_{\mathcal{G},K}^t.$$

The inequality holds with probability at least $1-\zeta$ because $\Pr(\mathcal{E}_{T,K}) \ge 1-\zeta$. □

Thus the local coefficients are collected into $a_{\mathcal{G},K}$ and $\gamma_{\mathcal{G},K}$, while the dependence on the communication graph and depth remains explicit through the two factors

$$Q_{\mathcal{G}}(K) = \|M_{\mathcal{G}}^K\|_{\mathsf{X}}, \qquad S_{\mathcal{G}}(K) = \sum_{q=0}^{K-1} \|M_{\mathcal{G}}^q\|_{\mathsf{X}}.$$

A.6 Propagation regimes and spectral attainability

This subsection records what follows from the affine comparison recursion and separates an upper-envelope statement from an attainability statement. The topology factors enter the one-step bound through the communication block before the horizon recursion is iterated. The resulting comparison recursion implies the usual stable, linear, and amplifying task-horizon regimes. A Perron-aligned construction shows that the spectral dependence of the communication block can occur within the nonnegative comparison class, following the Perron–Frobenius theory for nonnegative matrices (Seneta, 2006).

Task-horizon scaling. For fixed $(\mathcal{G}, K)$, the communication block first determines $Q_{\mathcal{G}}(K)$ and $S_{\mathcal{G}}(K)$. These two factors determine the one-step feedback and injection terms before the resulting map is repeated over the task horizon. In the compact notation of the main theorem, this repetition has three task-horizon regimes. If $a_{\mathcal{G},K} < 1$ and $\gamma_{\mathcal{G},K} > 0$, the scalar comparison envelope converges to $\gamma_{\mathcal{G},K}/(1-a_{\mathcal{G},K})$. If $a_{\mathcal{G},K} = 1$ and $\gamma_{\mathcal{G},K} > 0$, it is $\delta_0 + T\gamma_{\mathcal{G},K}$ and therefore grows linearly with T. If $a_{\mathcal{G},K} > 1$ and $\delta_0 + \gamma_{\mathcal{G},K} > 0$, it grows at rate $\Theta(a_{\mathcal{G},K}^T)$. These statements describe the scalar comparison envelope.

Proposition 2 (Perron-aligned spectral attainability)**.** *Let $\rho_{\mathcal{G}} := \rho(M_{\mathcal{G}})$. The nonnegative comparison class contains an instance with a nonnegative vector v, normalized by $\|v\|_{\mathsf{X}} = 1$, for which, at every task step,*

$$\|\Delta_t^{(K)}\|_{\mathsf{X}} = \rho_{\mathcal{G}}^K(c_0\delta_t + \bar{\xi}) + (b\delta_t + \bar{\varepsilon}) \sum_{q=0}^{K-1} \rho_{\mathcal{G}}^q.$$

The proposition concerns the communication comparison class and supports the dominant spectral rate.

Proof. By Perron–Frobenius, choose $v \succeq 0$ with $M_{\mathcal{G}}v = \rho_{\mathcal{G}}v$ and $\|v\|_{\mathsf{X}} = 1$. Choose the comparison instance with $W_t^{(k)} = M_{\mathcal{G}}$, $\Delta_t^{(0)} = (c_0\delta_t + \bar{\xi})v$, and $B_t^{(k)}\delta_t + \varepsilon_t^{(k)} = (b\delta_t + \bar{\varepsilon})v$ for every round. The one-round recursion is then attained with equality. After K rounds,

$$\Delta_t^{(K)} = \left[\rho_{\mathcal{G}}^K(c_0\delta_t + \bar{\xi}) + (b\delta_t + \bar{\varepsilon}) \sum_{q=0}^{K-1} \rho_{\mathcal{G}}^q\right] v.$$

Taking the X-norm proves the claim. The cases $\rho_{\mathcal{G}} < 1$, $\rho_{\mathcal{G}} = 1$, and $\rho_{\mathcal{G}} > 1$ give the corresponding bounded, linear, and exponential communication-depth rates whenever the injected term is nonzero. □

Taking equality in the action and macro-transition interfaces along this Perron-aligned instance gives the factor-resolved recursion

$$\delta_{t+1} = \Big[\mu_0 + \mu_a\lambda_c\Big(c_0\rho_{\mathcal{G}}^K + b\sum_{q=0}^{K-1}\rho_{\mathcal{G}}^q\Big)\Big]\delta_t + \varepsilon_0 + \mu_a\lambda_c\Big(\bar{\xi}\rho_{\mathcal{G}}^K + \bar{\varepsilon}\sum_{q=0}^{K-1}\rho_{\mathcal{G}}^q\Big).$$

This realizes the spectral dependence with separate transport and accumulation coefficients. The compact coefficients $a_{\mathcal{G},K}$ and $\gamma_{\mathcal{G},K}$ additionally use induced-norm bounds and the coefficient relaxation in Appendix A.5; their closed form remains an upper envelope.

Theorem 1 bounds deviations under the stated local propagation conditions. Appendix B.3 constructs a rollout satisfying these conditions in which a local perturbation propagates through context updates and removes task-relevant information from the transient input. This construction provides an explicit setting to which the endogenous recovery bound and persistent memory separation apply.

B Recovery Limits and Persistent Memory Sufficiency (Proofs for Section 5)

This appendix gives the proofs of the information-theoretic recovery limit and the persistent-memory separation in Section 5. The first proof uses the data-processing inequality and Fano's inequality (Cover & Thomas, 2005; Fano & Wintringham, 1961). The second proof constructs a distributed task whose target is available after aggregation, but whose task-relevant information is removed from the transient state by a local perturbation followed by communication and context updates. We then prove the zero-error persistent-memory construction and its state-cardinality lower bound. The final subsections establish decision identifiability and sufficient conditions for precondition recovery.

Theorem 2 gives the protocol-independent Bayes floor $\mathcal{F}_{\mathrm{EI}}^{(K)}(D) \geq \mathrm{err}(Y_D \mid X_D)$, together with an explicit Fano relaxation for any distribution on the finite target set $\mathcal{A}_R$, where $|\mathcal{A}_R| = R \geq 2$. On the constructed task, Theorem 3 attains this floor exactly at $1 - 1/R$; zero-error recovery requires and is achieved by R persistent states under the stated causal-writing and exact-retention conditions. Appendix B.6 applies decision identifiability to the auxiliary verdict V_t and proves exact precondition recovery under its evidence-validity, checking, and retrieval conditions.

B.1 Endogenous interaction and the information constraint

This subsection records the information restriction used by the endogenous- interaction regime. For a protocol $\pi \in \Pi_{\mathrm{EI}}^{(K)}$, let $Z_{D,\pi}^{1:K}$ denote its communication transcript after the decision-time transient input X_D. The protocol may transform, communicate, and combine information already present in X_D, but it has no access to additional task-relevant evidence. Thus,

$$I\big(Y_D; Z_{D,\pi}^{1:K} \mid X_D\big) = 0.$$

This conditional-information formulation is the Markov condition needed below; it does not require a particular communication graph or decoder.

Lemma 4 (Bayes floor under endogenous interaction)**.** *For every $K \geq 0$ and every $\pi \in \Pi_{\mathrm{EI}}^{(K)}$ satisfying $I(Y_D; Z_{D,\pi}^{1:K} \mid X_D) = 0$,*

$$\Pr[\widehat{Y}_{D,\pi} \neq Y_D] \geq \mathrm{err}(Y_D \mid X_D).$$

Consequently, $\mathcal{F}_{\mathrm{EI}}^{(K)}(D) \geq \mathrm{err}(Y_D \mid X_D)$.

Proof. The conditional-information assumption gives $\mathcal{L}(Y_D \mid X_D, Z_{D,\pi}^{1:K}) = \mathcal{L}(Y_D \mid X_D)$ almost surely. Therefore a decoder using $(X_D, Z_{D,\pi}^{1:K})$ has the same conditional target posterior as a decoder using X_D; randomization in the decoder can only average decisions under that posterior. The minimum error achievable after observing the transcript is consequently no smaller than the Bayes decision error based on X_D, which is $\mathrm{err}(Y_D \mid X_D)$. Taking the infimum over protocols proves the second claim. □

The following proof establishes the lower bound for every fixed communication depth K. It first applies data processing to the final prediction, then uses Fano's inequality to convert the retained mutual information into task error, and finally takes the infimum over admissible protocols.

Proof of Theorem 2. Fix $K \geq 0$ and $\pi \in \Pi_{\mathrm{EI}}^{(K)}$. Because the final prediction $\widehat{Y}_{D,\pi}$ is a possibly randomized post-processing of $(X_D, Z_{D,\pi}^{1:K})$, the data-processing inequality gives

$$I(Y_D; \widehat{Y}_{D,\pi}) \leq I\big(Y_D; X_D, Z_{D,\pi}^{1:K}\big).$$

The chain rule and the endogenous-interaction information constraint imply

$$\begin{aligned} I\big(Y_D; X_D, Z_{D,\pi}^{1:K}\big) &= I(Y_D; X_D) + I\big(Y_D; Z_{D,\pi}^{1:K} \mid X_D\big) \\ &= I(Y_D; X_D). \end{aligned}$$

Let $P_{e,\pi} := \Pr[\widehat{Y}_{D,\pi} \neq Y_D]$.

Fano's inequality gives

$$H(Y_D \mid \widehat{Y}_{D,\pi}) \leq h_2(P_{e,\pi}) + P_{e,\pi} \log_2(R-1),$$

where h_2 is binary entropy in bits. On the other hand, data processing implies

$$\begin{aligned} H(Y_D \mid \widehat{Y}_{D,\pi}) &\geq H\big(Y_D \mid X_D, Z_{D,\pi}^{1:K}\big) \\ &= H(Y_D) - I(Y_D; X_D). \end{aligned}$$

Combining the last two displays and using $h_2(P_{e,\pi}) \leq 1$ and $\log_2(R-1) \leq \log_2 R$ gives

$$P_{e,\pi} \geq \frac{H(Y_D) - I(Y_D; X_D) - 1}{\log_2 R}.$$

The probability of error is nonnegative, so the right-hand side can be replaced by its positive part. The inequality holds for every $\pi \in \Pi_{\mathrm{EI}}^{(K)}$, taking the infimum over π proves the Fano part of Equation equation 2.

To establish the intermediate Bayes-error bound, consider any measurable decision rule f based on X_D, and let $p_f = \Pr[f(X_D) \neq Y_D]$. Data processing and Fano's inequality give

$$H(Y_D) - I(Y_D; X_D) = H(Y_D \mid X_D) \leq H(Y_D \mid f(X_D)) \leq 1 + p_f \log_2 R.$$

Taking the infimum over all such rules yields

$$\mathrm{err}(Y_D \mid X_D) \geq \left[\frac{H(Y_D) - I(Y_D; X_D) - 1}{\log_2 R}\right]_+.$$

Combining this bound with Lemma 4 gives

$$\mathcal{F}_{\mathrm{EI}}^{(K)}(D) \geq \mathrm{err}(Y_D \mid X_D) \geq \left[\frac{H(Y_D) - I(Y_D; X_D) - 1}{\log_2 R}\right]_+.$$

□

For a finite target set, the Bayes error is zero if and only if the conditional target distribution assigns probability one to a single answer almost surely. This is equivalent to $H(Y_D \mid X_D) = 0$. Hence $H(Y_D \mid X_D) > 0$ implies $\mathrm{err}(Y_D \mid X_D) > 0$.

B.2 Distributed task and causal persistent state

This subsection constructs the task used in the separation theorem. The construction uses independent uniform shares over the cyclic group $\mathbb{Z}_R$. The sum of all shares is uniform, whereas every proper subset of shares is independent of the sum. This is the standard additive-share construction used to distribute information across agents.

Fix a bijection $\iota : \mathcal{A}_R \to \mathbb{Z}_R$. Draw $S_1, \ldots, S_N$ independently and uniformly from $\mathbb{Z}_R$, and set

$$U := \left(\sum_{i=1}^{N} S_i\right) \bmod R, \qquad Y_D := \iota^{-1}(U).$$

Uniform target and proper-subset independence. Conditioning on any $N-1$ shares leaves one independent uniform summand, so U, and therefore Y_D, is uniform. More generally, for every strict subset $J \subsetneq [N]$, choose $j \notin J$. Conditional on $(S_i)_{i\in J}$, the summand S_j remains independent and uniform, and addition modulo R is a bijection. Hence

$$Y_D \perp (S_i)_{i\in J} \qquad \text{for every } J \subsetneq [N].$$

This proves the distributional part of the task construction.

Causal writing before information loss. By the assumption in the theorem, all shares have been aggregated by $t_0 < D$. A designated writer can therefore compute U using only information available at t_0 and write

$$M_{t_0} := U, \qquad M_D = M_{t_0}.$$

The memory is thus written causally before the later transient-state loss.

B.3 An error propagation induced information loss

For the distributed task in Appendix B.2, we construct a rollout within the propagation class of Theorem 1. A local perturbation erases the aggregated target symbol at one agent, and subsequent message passing spreads this erasure through context replacement. We establish both that the rollout satisfies the local propagation conditions and that its final transient input is independent of the target.

Lemma 5 (Propagation-induced transient information loss)**.** *For the distributed-share task above, there is a finite-state pre-decision rollout satisfying the local propagation conditions of Section 4 such that the complete transient input at time D satisfies $X_D \perp Y_D$.*

Proof. To construct an admissible post-aggregation information-loss rollout, consider a directed star with center agent 1 and edges $1 \to i$ for $i \neq 1$. At t_0, before the perturbation, the actual and ideal contexts of every agent contain the aggregated symbol U. At a subsequent communication round, a local perturbation in the center's context update replaces its actual aggregated symbol by a public filler symbol $\perp$, while the ideal context continues to contain U. The center sends its actual context to each receiver. Each receiver updates its context from the received message, so after this round every actual context contains $\perp$. The private shares are not retained as separate transient inputs after aggregation; subsequent observations and the decision query are public filler independent of U.

This is a finite-state instance of the local conditions in Section 4: the message map copies the sender context, the receiver update is a local context update, and the center perturbation appears as a bounded residual term. All maps on the finite context space have finite local sensitivity constants. The resulting semantic drift is transported from the center to the receivers along the star edges and is covered by the one-round bound in Appendix A.2. The the subsequent macro-state and context feedback can be taken to use the public filler observation. Thus the rollout is an admissible member of the Section 4 comparison class.

The information loss follows from the many-to-one context update: every possible aggregated symbol U is replaced by the same public filler $\perp$. Message passing propagates this replacement to all agents. Since the private shares are no longer retained and the remaining observations and query are independent of U, the complete transient input X_D is the same for every target value. Hence,

$$X_D \perp Y_D.$$

□

Before the perturbation, X_{t_0} contains U, which is causally retained as $M_D = U$. Since $Y_D = \iota^{-1}(U)$, the construction gives

$$I(Y_D; X_{t_0}) = H(Y_D), \qquad I(Y_D; X_D) = 0, \qquad H(Y_D \mid X_D, M_D) = 0.$$

Thus, an admissible propagation process can remove all target information from the transient input while leaving it recoverable from a persistent state written before the loss. Theorem 2 characterizes the resulting recovery limit under endogenous interaction, and Theorem 3 establishes the exact separation achieved by persistent memory.

B.4 Proof of the Theorem 3

This subsection proves the two parts of Theorem 3. The first part uses the constructed rollout to show exact random-guessing error under endogenous interaction. The second part decodes the causally written state $M_D = U$ and then proves that fewer than R deterministic memory states cannot support zero-error recovery. The latter step uses the standard entropy bound $H(V) \le \log_2 |\mathcal{V}|$ for a finite-valued variable, as in classical exact-recovery lower bounds (Karnin et al., 1983).

Proof of Theorem 3. Use the distributed task from Appendix B.2 and the admissible propagation-induced information-loss rollout from Lemma 5. The construction gives $Y_D \sim \mathrm{Unif}(\mathcal{A}_R)$, $Y_D \perp (S_i)_{i\in J}$ for every proper subset J, and $X_D \perp Y_D$.

Endogenous interaction. Fix $K \ge 0$ and $\pi \in \Pi_{\mathrm{EI}}^{(K)}$. The endogenous-interaction constraint and $X_D \perp Y_D$ imply

$$I\big(Y_D; X_D, Z_{D,\pi}^{1:K}\big) = I(Y_D; X_D) + I\big(Y_D; Z_{D,\pi}^{1:K} \mid X_D\big) = 0.$$

By data processing, the final prediction is independent of Y_D. Therefore, for any possibly randomized decoder,

$$\Pr[\widehat{Y}_{D,\pi} = Y_D] = \sum_{a\in\mathcal{A}_R} \Pr[\widehat{Y}_{D,\pi} = a] \Pr[Y_D = a] = \frac{1}{R}.$$

Thus every admissible protocol has error $1 - 1/R$, and taking the infimum gives

$$\mathcal{F}_{\mathrm{EI}}^{(K)}(D) = \mathrm{err}(Y_D \mid X_D) = 1 - \frac{1}{R}.$$

Zero-error persistent memory. The writer stores $M_D = U$. The decoder

$$f(x, m) := \iota^{-1}(m)$$

then satisfies $f(X_D, M_D) = Y_D$ almost surely, so

$$\mathrm{err}(Y_D \mid X_D, M_D) = 0.$$

Because U is uniform on $\mathbb{Z}_R$, the construction uses exactly $|\mathcal{M}| = R$ possible memory states.

Minimum number of persistent states. Let $\widetilde{M}_D$ be any finite-valued deterministic persistent state that permits zero-error recovery. Then

$$H(Y_D \mid X_D, \widetilde{M}_D) = 0.$$

Since $X_D \perp Y_D$ and Y_D is uniform,

$$\begin{aligned}
\log_2 R &= H(Y_D \mid X_D) \\
&= I(Y_D; \widetilde{M}_D \mid X_D) + H(Y_D \mid X_D, \widetilde{M}_D) \\
&\le H(\widetilde{M}_D \mid X_D) \\
&\le \log_2 |\widetilde{\mathcal{M}}|.
\end{aligned}$$

Hence $|\widetilde{\mathcal{M}}| \ge R$. The construction with R states is therefore both sufficient and minimal. □

B.5 Proof of memory decision identifiability

The next proposition provides a deterministic characterization of exact recovery over a finite history set. The characterization follows from the induced equivalence classes of decoder inputs: exact recovery is possible if and only if each class contains a unique target value. Specifically, conflicting targets within the same decoder-input class make exact recovery impossible, whereas a constant target assignment over every class defines an exact decoder.

Proof of Proposition 1. For necessity, suppose that there are histories h, h' such that

$$X_t(h) = X_t(h'), \qquad Y_t(h) \neq Y_t(h'), \qquad M_t(h) = M_t(h').$$

Both histories give the decoder the same pair (X_t, M_t), so any deterministic decoder must return the same value on both. It cannot equal both different targets, contradicting exact recovery.

For sufficiency, partition the feasible histories according to the pair $(X_t(h), M_t(h))$. The stated condition implies that $Y_t(h)$ is constant on each such class: if two members had different targets, the condition would force different memory states despite having the same transient input. Define $g(x, m)$ to be this common target on every nonempty class, and define it arbitrarily elsewhere. Then

$$g(X_t(h), M_t(h)) = Y_t(h) \qquad \text{for every } h \in \mathcal{H}_t,$$

which proves the proposition. □

Under a distribution with positive probability on every feasible history, the same condition is equivalent to

$$H(Y_t \mid X_t, M_t) = 0, \qquad I(Y_t; M_t \mid X_t) = H(Y_t \mid X_t).$$

Exact recovery requires memory to resolve the target distinctions left unresolved by the transient input. This condition characterizes information sufficiency, while permitting different memory representations and storage costs.

B.6 Sufficient conditions for precondition recovery

The following condition connects Proposition 1 to the selective insertion procedure in Appendix C.1. Let E_t and M_t contain current and retained evidence, respectively. Include the candidate command and its public query parameters in the decision input X_t, so that $P_t = \mathrm{Pre}(\tilde{a}_t)$ and E_t are fixed functions of X_t. Fix the selection and checking rules as functions of their displayed inputs. Let $V_t \in \{\mathrm{true}, \mathrm{false}\}$ be the correct auxiliary verdict for the candidate, while Y_t retains its task-target meaning. A sufficient witness supports every declared precondition or refutes at least one, with the required source dependencies and validity checks. Unresolved conflicts cannot establish support or refutation.

Proposition 3 (Sufficient condition for exact precondition recovery)**.** *Assume the following conditions on a finite feasible-history class:*

1. *all fact and relation judgments accepted by* Check *are correct and valid for the current query;*
2. *in every feasible history, $E_t \cup M_t$ contains a sufficient witness, and* Check *recognizes sufficient witnesses after a complete dependency lookup;*
3. *if the current check is undetermined, the selector based on (X_t, M_t) returns $M_t^{\mathrm{inj}} = W_t \setminus E_t$ for one such witness $W_t \subseteq E_t \cup M_t$, preserving its validity; otherwise $M_t^{\mathrm{inj}} = \varnothing$.*

Then

$$\mathrm{Check}(P_t, E_t \cup M_t^{\mathrm{inj}}) = V_t \quad \textit{for every feasible history},$$

and the pair (X_t, M_t) satisfies the separation condition in Proposition 1 instantiated with V_t as the queried variable. Consequently, $H(V_t \mid X_t, M_t) = 0$ under any distribution supported on the feasible histories.

Proof. If the current check is determined, condition 1 makes its verdict correct. Otherwise, conditions 2–3 ensure that $E_t \cup M_t^{\mathrm{inj}}$ contains a recognized sufficient witness: all required support for a true verdict, or a valid refutation for a false verdict. Condition 1 makes the resulting verdict equal to Y_t.

The query, current evidence, selection, and check depend only on (X_t, M_t), so their composition defines a decoder that is correct on every feasible history. Two histories with the same pair (X_t, M_t) therefore have the same correct verdict. This is the separation condition in Proposition 1 for V_t, and exact decoding also gives $H(V_t \mid X_t, M_t) = 0$ under any distribution on this history class. □

This proposition provides a sufficient design criterion for **RECOVER**. Under the stated conditions on evidence validity, checking, and retrieval, retaining a sufficient witness and injecting its missing records determines the candidate's precondition verdict. This motivates preserving evidence for subsequent queries and supplying it when the current input leaves the verdict undetermined. The guarantee concerns precondition recovery. Token and storage efficiency are evaluated empirically.

C What to Remember and When to Inject: Procedure and Case Studies

We instantiate the main text's memory rule through a shared evidence procedure and five trial-level examples. The examples are re[resented as case studies drawn from actual runs of RECOVER on different benchmarks.

C.1 Memory procedure

RECOVER maintains observed facts and supplies them when they resolve an undetermined precondition check. The notation $\mathrm{Extract}$, Update, Pre, and Check names operations within this single procedure. The same checking and injection rule is used across datasets. An LLM derives precondition templates once from each dataset's few-shot examples; the templates then remain fixed across task instances. Pre fills in a template using the candidate action's operator, arguments, and public query parameters.

M_t stores valid facts from task observations together with their sources. E_t is the evidence extracted from the complete transient input X_t before memory insertion. For candidate action $\tilde{a}_t$, let $P_t = \mathrm{Pre}(\tilde{a}_t)$ and let V_t denote the true verdict of these preconditions.

After executing an action, $\mathrm{Extract}$ records facts supported by the resulting observation. Update replaces superseded facts, removes invalidated facts, and retains the remaining records:

$$M_{t+1} = \mathrm{Update}(M_t, \mathrm{Extract}_{\mathrm{Pre}}(a_t, o_{t+1})) \,. \tag{6}$$

Facts relevant to the fixed templates can be stored before a later candidate requires them.

Check returns true when all preconditions are supported, false when at least one is refuted by unconflicted evidence, and $\perp$ otherwise. For an unknown current check, RECOVER reads valid stored facts relevant to P_t and applies the same check to the augmented evidence:

$$\mathrm{Insert}_t = \mathbf{1}\big[\mathrm{Check}(\mathrm{Pre}(\tilde{a}_t), E_t) = \perp \wedge \mathrm{Check}(\mathrm{Pre}(\tilde{a}_t), E_t \cup M_t) \neq \perp\big]. \tag{7}$$

Both checks use the same preconditions and verification rules. If the augmented check is determined, let W be the supporting records used for that verdict. RECOVER injects $B = W \setminus E_t$, including its sources. $\mathrm{Revise}_{\mathrm{MAS}}$ then revises the original proposal with B added to the underlying MAS input.

If no memory is injected, the original proposal is executed unchanged, including when the current check returns false. Each insertion invokes one additional agent call to revise the proposal, with its tokens and latency included in the reported costs. The benchmark-specific evidence checks are described below. Evaluation rewards and reference answers are not inputs to the procedure.

C.2 Case Studies and Benchmark Instantiations

The rightmost column uses $(\mathrm{Check}(P, E), \mathrm{Check}(P, E \cup M)) \Rightarrow \mathrm{Insert}$. Checks return True, False, or $\perp$ (unknown). Each listed E specifies the recipient view for that readout. All unmentioned evidence relevant to the check is absent. Memory retains only facts whose scope remains valid. A dash marks a write-only event. Observations in the tables are shortened to the relevant content, and unshown intermediate events are accounted for when determining validity.

Algorithm 1 Task-local evidence update and selective insertion

Require: Task q, fixed dataset-level templates, initial observation o_0

$M \leftarrow \mathrm{Update}(\varnothing, \mathrm{Extract}_{\mathrm{Pre}}(\varnothing, o_0))$
while the task is active **do**
  Receive X; extract current evidence E from X
  MAS proposes $\tilde{a}$ from X; $a \leftarrow \tilde{a}$
  Instantiate $P \leftarrow \mathrm{Pre}(\tilde{a})$ using q
  if P is supported and $\mathrm{Check}(P, E) = \bot$ **then**
    Read valid stored facts relevant to P, including conflicting evidence
    $(c, W) \leftarrow \mathrm{Check}(P, E \cup M)$ using those facts
    if $c \neq \bot$ **then**
      $B \leftarrow W \setminus E$ with supporting sources
      if $B \neq \varnothing$ **then**
        $a \leftarrow \mathrm{MAS}(X, \tilde{a}, B)$
      end if
    end if
  end if
  Execute a and receive observation o
  $M \leftarrow \mathrm{Update}(M, \mathrm{Extract}_{\mathrm{Pre}}(a, o))$
end while
Clear task memory

C.2.1 ALFWORLD: POSSESSION AND LOCATION

The source is a recorded lettuce-cooling task. For placing lettuce 1 on countertop 3, the declared conditions are

$$\mathrm{Pre}(\mathrm{Place}(\text{lettuce 1}, \text{countertop 3})) = \{\mathrm{Held}(\text{lettuce 1}) = \mathrm{True},\ \mathrm{At}(\mathrm{agent}) = \text{countertop 3}\}.$$

Location and possession acknowledgements establish the values; visibility records are scoped to the observed context.

Table 2: ALFWorld: the same retained facts can refute or support placement.

Step	Action and observation	Memory after update	Candidate, E, and insertion
9	Take(lettuce 1, countertop 2). Environment confirms pickup.	$\mathrm{Held}(\text{lettuce 1}) = \mathrm{True}$; $\mathrm{LocatedAt}(\text{lettuce 1}) = \mathrm{inventory}$.	Write confirmed possession; no candidate fact is created.
11	Navigate(fridge 1). Arrival at the fridge is observed.	$\mathrm{At}(\mathrm{agent}) = \text{fridge 1}$; possession remains true.	Place(lettuce 1, countertop 1); E lacks location evidence. $(\bot, \mathrm{False}) \Rightarrow 1$: inject the fridge location.
16	Navigate(countertop 3). The countertop is observed.	$\mathrm{At}(\mathrm{agent}) = \text{countertop 3}$; possession remains true.	Place(lettuce 1, countertop 3); E has location, not possession. $(\bot, \mathrm{True}) \Rightarrow 1$: inject the pickup evidence.
17	Place(lettuce 1, countertop 3). Environment confirms placement.	$\mathrm{Held}(\text{lettuce 1}) = \mathrm{False}$; $\mathrm{LocatedAt}(\text{lettuce 1}) = \text{countertop 3}$.	Old held value is superseded. The task terminates and memory is cleared.

C.2.2 SCIENCEWORLD: REACHABILITY WITHIN A ROOM

The candidate is navigation to the hallway. Its declared condition is

$$\mathrm{Pre}(\mathrm{Navigate}(\textbf{hallway})) = \{\mathrm{Reachable}(\textbf{hallway}) = \mathrm{True}\}.$$

A closed-door response and an opening acknowledgement support opposite values at successive revisions. Reachability and door visibility expire when the agent leaves the room context in which they were established. Table 3 is a worked closed-door interaction.

Table 3: ScienceWorld: readout changes as the observation and its scope change.

Step	Action and observation	Memory after update	Candidate, E, and insertion
1	$\mathrm{Navigate}(\textbf{hallway})$. Response: cannot pass; the door is closed.	$\mathrm{Reachable}(\textbf{hallway}) = \mathrm{False}$, with rejection evidence.	For a later navigation proposal with this response absent from E: $(\bot, \mathrm{False}) \Rightarrow 1$.
2	$\mathrm{LookAround}()$. The hallway door is explicitly visible.	$\mathrm{DoorVisible}(\textbf{hallway}) = \mathrm{True}$; reachability remains false.	For $\mathrm{OpenDoor}(\textbf{hallway})$, E contains door visibility: $(\mathrm{True}, \mathrm{True}) \Rightarrow 0$.
3	$\mathrm{OpenDoor}(\textbf{hallway})$. Response: the door is now open.	$\mathrm{Reachable}(\textbf{hallway}) = \mathrm{True}$ replaces the false value.	For navigation with the acknowledgement still in E: $(\mathrm{True}, \mathrm{True}) \Rightarrow 0$.
4	A later relocation changes the room context.	Old local reachability and visibility support expire.	For the old context's query with no fresh evidence: $(\bot, \bot) \Rightarrow 0$.

C.2.3 FEVER: RETAINING THE SOURCE PACKET FOR A CLAIM

Recorded task 232 verifies the claim *Global warming threatens food security*. Memory keeps source sentences and the dependencies of accepted relation findings. The candidate's evidence obligation is

$$\mathrm{Pre}(\mathrm{Finish}(\mathrm{SUPPORTS})) = \{\mathrm{Entails}(\textbf{Global warming threatens food security}) = \mathrm{True}\}.$$

Table 4: FEVER: the same source packet is used only when absent from the readout view.

Step	Action and observation	Memory after update	Candidate, E, and insertion
2, 4	Search for global warming and food security. Responses define the topics.	Retain the source sentences and page identifiers.	$\tilde{a} = \mathrm{Finish}(\mathrm{SUPPORTS})$; neither view establishes the relation: $(\bot, \bot) \Rightarrow 0$.
6	Search for their connection. A source sentence reports stresses to food-producing systems.	Retain the sentence and any accepted entailment finding with its source support.	If E contains the accepted packet: $(\mathrm{True}, \mathrm{True}) \Rightarrow 0$.
7	$\mathrm{Thought}$; environment response: `OK.`	No new world fact; the source packet remains valid.	Before the recorded final proposal, set E to the claim and `OK.` only: $(\bot, \mathrm{True}) \Rightarrow 1$. Inject the source packet.

C.2.4 HOTPOTQA: RETAINING THE FIRST HOP OF A JOIN

Our HotpotQA adapter extracts source-backed relations from retrieval responses and evaluates the candidate answer by joining their entity bindings. For the question asking which county contains the village where *Guitars for Wounded Warriors* was recorded, the candidate is $\mathrm{Finish}(\text{Ulster County})$. Its Pre requires a source-backed path from the album's recording village to Ulster County. The two retrievals supply

$$\mathrm{RecordedIn}(\text{Guitars for Wounded Warriors}, \text{New Paltz}),$$
$$\mathrm{CountyOf}(\text{New Paltz}, \text{Ulster County}).$$

The adapter retains each relation with its source span. Check joins on New Paltz and tests whether the resulting county matches the candidate answer. When the first passage is absent from E_t, memory supplies that passage; when both are visible, no insertion is needed. Missing hops or ambiguous bindings remain unknown. The join is computed for the current candidate, and a source revision invalidates any dependent finding.

Table 5: HotpotQA: memory supplies the missing operand of the same two-hop query.

Step	Action and observation	Memory after update	Candidate, E, and insertion
1	Search: Guitars for Wounded Warriors. The album was recorded in New Paltz.	$\mathrm{RecordedIn}$: Guitars for Wounded Warriors $\rightarrow$ New Paltz, with the first passage.	For the candidate answer Ulster County, no county evidence is yet available: $(\bot, \bot) \Rightarrow 0$.
2	$\mathrm{Search}(\text{New Paltz})$. New Paltz is in Ulster County.	Add $\mathrm{CountyOf}$: New Paltz $\rightarrow$ Ulster County; retain the first-hop tuple.	$\mathrm{Finish}(\text{Ulster County})$; E contains only the county passage: $(\bot, \mathrm{True}) \Rightarrow 1$. Inject the recording-location passage.
–	Alternative readout of the same two responses.	Both source-backed tuples remain unchanged.	E contains both passages: $(\mathrm{True}, \mathrm{True}) \Rightarrow 0$.

C.2.5 OFFICEBENCH: APPLICATION STATE AND WORKBOOK EVIDENCE

Recorded task 1-13/0 asks whether a student's prior courses include CS101, CS102, Math151, and Math152. The agent first requests `student_courses.xlsx`; the directory later identifies `Thomas_Robinson_Course_List.xlsx`. For ReadFile on the identified workbook, Pre requires

$$\text{ActiveApp} = \text{Excel},$$
$$\text{Exists}(\texttt{Thomas_Robinson_Course_List.xlsx}) = \text{True}.$$

For Finish(Yes), Pre requires source cells establishing that CS101, CS102, Math151, and Math152 are recorded in that workbook. Course IDs are normalized for case and whitespace. App switches update the app key; file edits invalidate affected cells and derived memberships. Shell listing does not itself switch the active office application. Evaluator output is not a memory input.

Table 6: OfficeBench: one rule checks both tool state and the answer's table evidence.

Step	Action and observation	Memory after update	Candidate, E, and insertion
1	ReadFile: `student_courses.xlsx`. Response: switch to Excel first.	Record that the Excel-active condition is false.	For another read with that error still in E: $(\text{False}, \text{False}) \Rightarrow 0$.
2, 3	SwitchApp(Excel) succeeds; ReadFile on `student_courses.xlsx` reports file not found.	$\text{ActiveApp} = \text{Excel}$; Exists for `student_courses.xlsx` is False.	For another read of `student_courses.xlsx` with its error visible: $(\text{False}, \text{False}) \Rightarrow 0$.
4	ListDirectory(). The response lists `Thomas_Robinson_Course_List.xlsx`.	Add Exists = True for `Thomas_Robinson_Course_List.xlsx`; the app state persists.	ReadFile: `Thomas_Robinson_Course_List.xlsx`; E has the listing but lacks app evidence: $(\perp, \text{True}) \Rightarrow 1$. Inject the switch acknowledgement.
5	ReadFile: `Thomas_Robinson_Course_List.xlsx`. Rows 2, 5, 6, 9 contain the four required course IDs.	Store the source cells and their file version; they support the four Recorded predicates.	$\tilde{a} = \text{Finish(Yes)}$; all four rows in E: $(\text{True}, \text{True}) \Rightarrow 0$. Only row 9 absent: $(\perp, \text{True}) \Rightarrow 1$.

The five examples use the same write and readout rules. Their source observations determine what can be retained, and the candidate determines which evidence matters at that decision. Under sound extraction and checking, memory completes a distinction absent from the specified recipient view.

D EXPERIMENTAL VALIDATION OF PROPOSED THEOREMS

D.1 EXPERIMENTAL SETUP AND CALIBRATION

Experimental framework and calibration. We organize the empirical analysis around the communication operator, the action–state interfaces, and their composition over task time. The communication gains are calibrated from paired generated messages using the one-hop procedure detailed below. The local-interface experiment separately estimates λ_c and μ_a from six calibration tasks and holds them fixed on 24 evaluation tasks. Each coefficient is defined with respect to its message representation, discrepancy metric, and reference process. Within a composed comparison, these representations and the reference process must be compatible across the communication, action, and state interfaces. The following protocols identify the intervention, reference, and measured quantity for each component of the theorem.

Figures 3–5 examine three components of Theorem 1: the local coupling from terminal semantic drift to actions and environment states, the persistence of deviation across task steps, and the scale of the two communication propagation factors. We use Qwen3-14B with a configurable communication-graph MAS on ALFWorld unseen-validation tasks and ScienceWorld test tasks.

Persistent memory is disabled. The following protocols define the intervention, reference, and measured quantity for each component.

Representations and reference executions. Each protocol fixes a representation for the quantities it compares. In the communication experiment, agents exchange natural-language messages during task execution. The semantic-drift measurement associated with agent i's generated message following factual-error injection. Each comparison uses its fixed representation and discrepancy measure to interpret the residuals and norm bounds. Starred quantities denote the designated unperturbed reference for the same task. Golden commands construct the local checkpoints and initial proposals. The local reference action is the zero-dose model output. The paired trajectory comparison shares the prefix before intervention and then follows each branch's recorded interaction. The general trajectory bound retains its initial-state term δ_{t_0}. The matched-checkpoint experiment instantiates the case $\delta_t = 0$.

For calibration task r, take the largest source-retention ratio and source-to-neighbor ratio over its three-graph bundle:

$$d_r = \max_{\mathcal{G}} \frac{\Delta_j^{(1)}}{\Delta_j^{(0)}}, \qquad g_r = \max_{\mathcal{G}} \max_{i:A_{ij}=1} \frac{\Delta_i^{(1)}}{\Delta_j^{(0)}}.$$

The source denominator is positive in each calibration comparison. Taking the maxima over calibration tasks gives

$$\widehat{d} = \max_r d_r = 0.83258305199, \qquad \widehat{g} = \max_r g_r = 0.90537521881. \tag{8}$$

These coefficients use calibration records alone and define the empirical operator $\widehat{M}_{\mathcal{G}} = \widehat{d}I + \widehat{g}A_{\mathcal{G}}$.

D.2 Figure 3: local micro–macro coupling

Protocol and measured quantities. We use six ALFWorld tasks for calibration and 24 separate tasks for evaluation, with one calibration task and four evaluation tasks from each of six task families. Three checkpoints per task yield 18 calibration and 72 evaluation checkpoints. These tasks are distinct from the 12 tasks used in the communication-factor study. The evaluation uses $N \in \{4, 5, 6\}$ and 96 entry- or injection-origin profiles drawn from 48 graph–depth configurations, as summarized in Table 11. At a checkpoint, all solver proposals initially recommend the golden command. We replace $m = 0, \ldots, N$ proposals with a fixed incorrect but admissible command. The replacement order is determined by an observed terminal communication profile. Every assigned profile is evaluated over all replacement levels.

Entry and injection orderings. An ordering specifies which agents' proposals are replaced first as m increases. The *entry ordering* uses the terminal drift profile produced by one unit injected at the entrance of the communication block, followed by K updates without further injection. The *injection ordering* uses the terminal profile produced from zero initial drift with one unit injected after each of the K updates. These profiles originate from the transport and accumulation channels of Figure 5, respectively. For either profile, agents are ranked by decreasing observed terminal drift mass, with ties broken by agent identifier. At dose m/N, we replace the proposals of the first m agents in that ranking. Proposals are presented to the decision agent in the same fixed agent order. Thus the two orderings can select different agent subsets at an intermediate dose while sharing the checkpoint, wrong command, input discrepancy, and decoder. Both use the same proposal bundle at $m = 0$ and $m = N$. Figures 9 and 10 show the two local response studies separately.

Configuration assignment and sample counts. Each of the 24 evaluation tasks contributes one checkpoint for each $N \in \{4, 5, 6\}$, giving 24 checkpoint assignments per size and ordering. Source profiles are assigned cyclically after sorting their graph–depth pairs by K and topology identifier. For $N = 4, 6$, the 12 available pairs are each used by two tasks. For $N = 5$, 24 different pairs are used: all ten graphs at $K = 1, 2$ and graphs M1, M10, M2, and M3 at $K = 3$. This yields $12 + 24 + 12 = 48$ local source configurations and $2 \times 48 = 96$ ordering-specific profiles. The source archive includes $K = 1$, whereas Figure 5 displays the communication conditions with $K \geq 2$. At each size, the local record count is $24 \times 2 \times (N + 1)$: 24 task checkpoints, two orderings, and $N + 1$ replacement levels. These counts are 240, 288, and 336, totaling 864 decision records.

Input, action, and state metrics. At a fixed checkpoint q, let a_q^{gold} and a_q^{wrong} be the two commands used to construct the proposals. The terminal input domain is $\mathcal{C}_{q,N} = \{a_q^{\mathrm{gold}}, a_q^{\mathrm{wrong}}\}^N$, with the task context and proposal presentation order held fixed. For a bundle in this domain, let $b_i \in \{0,1\}$ indicate replacement and define $c_i = b_i/N$. Relative to the zero-dose reference, we instantiate $\|\cdot\|_{\mathsf{X}}$ as $\|\cdot\|_1$ on these coordinates. Thus the local terminal discrepancy is the exact representation distance

$$u_t := \|\Delta_t^{(K)}\|_1 = \sum_{i=1}^{N} |c_i - c_i^{(0)}| = m/N.$$

A Qwen3-14B decision agent receives the task, episode context, admissible commands, and terminal proposals, and returns one executable command. The action canonicalizer takes the first output line, removes command prefixes and list markers, normalizes case and whitespace, and removes trailing full stops. Empty or thought-only outputs map to `NOOP`. The parser represents a recognized command by its verb and up to two arguments; absent arguments are denoted by $\varnothing$. For an unmatched command, it retains the first token as the verb and the normalized command as its first argument. Writing this three-slot frame as $F(a)$, define

$$I(a,a^*) = \{j : F_j(a) \neq \varnothing \text{ or } F_j(a^*) \neq \varnothing\}, \qquad c_{\mathcal{A}}(a,a^*) = \frac{\sum_{j\in I(a,a^*)} \mathbf{1}\{F_j(a) \neq F_j(a^*)\}}{|I(a,a^*)|}.$$

The empty-active-set convention is zero. Missing slots therefore count as mismatches when present in the other action. The recorded discrepancy is $e_t^a = c_{\mathcal{A}}(a_t, a_t^*)$, relative to the matched zero-dose model action. We replay both actions from the same factual checkpoint, giving $\delta_t = 0$, and measure their next-state discrepancy as $\delta_{t+1} = \frac{1}{4}\big(d_{\mathrm{location}} + d_{\mathrm{inventory}} + d_{\mathrm{attribute}} + d_{\mathrm{goal}}\big)$.

The first three terms are Jaccard distances over object-location facts, inventory, and state attributes. The last is the absolute difference in normalized goal progress. For sets U, V, the implemented Jaccard distance is $1 - |U \cap V|/|U \cup V|$, with value zero when both are empty. Both e_t^a and δ_{t+1} lie in $[0,1]$. The evaluation contains 864 decision records and 140 distinct checkpoint–action replays. Within each N, intervention levels share checkpoints. Across N, task identities are shared and checkpoints differ.

Action and state interfaces to the theoretical quantities. For a deterministic canonical action map, its action law is the point mass D_a. The transport discrepancy induced by the declared action cost satisfies

$$\mathcal{T}_{c_{\mathcal{A}}}(\mathsf{D}_a, \mathsf{D}_{a^*}) = c_{\mathcal{A}}(a,a^*), \qquad \mathcal{T}_c(P,Q) := \inf_{\kappa\in\Pi(P,Q)} \mathbb{E}_\kappa[c(A,A^*)].$$

This gives an exact action-law interpretation of the local cost for deterministic decoding. For random execution, the recorded action pair instead supplies a realization of the cost under its actual pairing κ, whose expectation is $\mathbb{E}_\kappa c(A,A^*)$. The marginal transport discrepancy is bounded above by this coupled expectation. The reported record-level statistics concern the realized paired costs under the recorded inference settings.

The measured state distance also has an exact norm representation. Let $\mathcal{S}$ be the space of factual-state tuples consisting of the three fact sets and normalized goal progress, and let $D_{\mathcal{S}}$ be their four-component distance above. Define the fixed representation $\psi(s) = (D_{\mathcal{S}}(s,z))_{z\in\mathcal{S}}$ in $\ell_\infty(\mathcal{S})$. Then

$$\|\psi(s) - \psi(s^*)\|_\infty = D_{\mathcal{S}}(s,s^*).$$

The triangle inequality gives the upper bound, and the coordinate $z = s$ attains it. Consequently, the existing state measurement realizes the theorem's norm notation on represented factual states. The experiment computes $D_{\mathcal{S}}$ directly; the representation establishes the mathematical correspondence without changing any measured distance.

Independent calibration and fixed comparisons. The retrospective interface analysis separates calibration and evaluation by task: every checkpoint, dose, and proposal ordering from one task stays in the same partition. The two partitions have no shared task, checkpoint, or environment task file. All 216 calibration records are used to estimate one common pair of coefficients. Writing u_r, e_r^a, and y_r for the input, action, and next-state discrepancies of record r, we set

$$\lambda_c = \max_{r\in\mathcal{C}:\, u_r>0} \frac{e_r^a}{u_r} = 2, \qquad \mu_a = \max_{r\in\mathcal{C}:\, e_r^a>0} \frac{y_r}{e_r^a} \approx 0.3922413793,$$

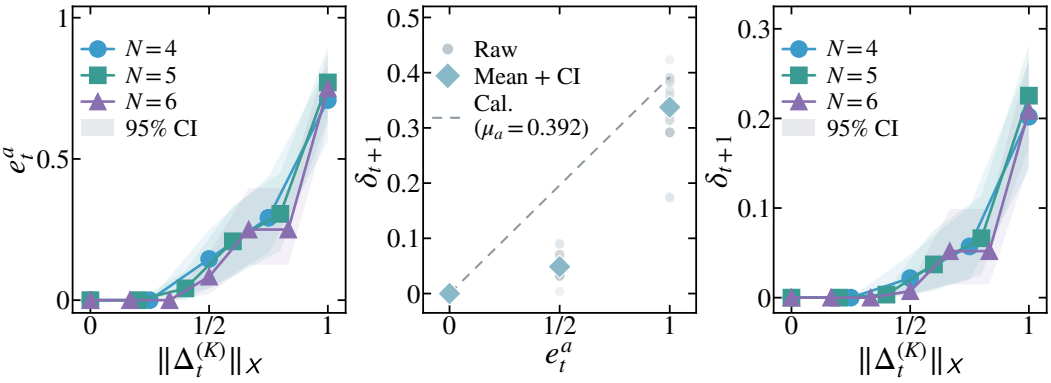

Figure 9: **Local micro–macro coupling under the entry ordering.** Proposals are replaced in decreasing order of the terminal drift produced by a single entry injection. Left and right panels show action and next-state responses to the replacement fraction m/N across $N = 4, 5, 6$. The center shows the action–state relation and the shared calibration line $\mu_a e_t^a$. Shading and bars show 95% task-bootstrap intervals.

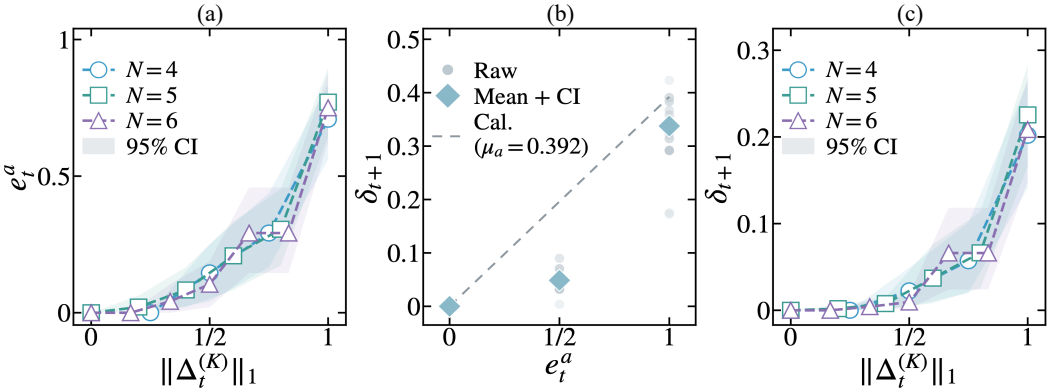

Figure 10: **Local micro–macro coupling under the injection ordering.** Proposals are replaced in decreasing order of the terminal drift produced by unit injections after every communication update. Panels and metrics match Figure 9. The checkpoint, dose sweep, decoder, and calibrated coefficient are shared across the two orderings. Shading and bars show 95% task-bootstrap intervals.

with $\bar{\nu} = \bar{\eta} = 0$ and separate checks of the zero-input cases. These coefficients are applied unchanged to every evaluation task, size, profile, and dose. Since the starting states match, the state term $\mu_s \delta_t$ vanishes. The complete local comparisons are therefore

$$B_r^a = 2u_r, \qquad B_r^s = \mu_a e_r^a, \qquad B_r^{\mathrm{comp}} = 2\mu_a u_r.$$

The line labeled Cal. in Figure 3(b) uses the fixed μ_a, rounded to 0.392 in the legend.

Interpreting the two ordering panels. Entry and injection orderings change the locations of proposal disagreements at a fixed replacement fraction. Applying the same calibration coefficients to both orderings tests the local comparisons across these two spatial patterns. The composed comparison $B^{\mathrm{comp}} = \lambda_c \mu_a u$ uses the input discrepancy, whereas $B^s = \mu_a e^a$ additionally uses the realized action discrepancy. Their signed gaps decompose as

$$B^{\mathrm{comp}} - \delta_{t+1} = \mu_a(\lambda_c u - e^a) + (\mu_a e^a - \delta_{t+1}).$$

The two terms locate the margin introduced at the decision and environment interfaces. This explains why composing the interfaces can give greater coverage together with greater slack, while the action-conditioned comparison provides a more resolved description after the paired actions are available.

Coverage, slack, and uncertainty. For a comparison $y_r \le B_r$, we report empirical coverage, positive slack, and exceedance:

$$\widehat{C} = \frac{1}{R}\sum_{r=1}^{R} \mathbf{1}\{y_r \le B_r\}, \qquad s_r = (B_r - y_r)_+, \qquad v_r = (y_r - B_r)_+.$$

The known unit metric range fixes $s_{\mathrm{ref}} = 1$, so these values retain their original units. We also report the fraction of tasks for which all sampled comparisons satisfy the bound. Confidence intervals resample whole tasks, preserving their checkpoints, doses, orderings, and repeated records. All tabulated intervals use 10,000 task-bootstrap resamples and are pointwise percentile 95% intervals. Correlations use the complete records with task-cluster resampling; per-task summaries and results by N, topology, depth, and configuration accompany the aggregate estimates. The confidence bands in Figure 3 use 5,000 task resamples. Record coverage and complete-task coverage describe the sampled comparisons. In the theorem, $\mathcal{E}_{T,K}$ is the event on which all primitive inequalities hold simultaneously along the specified rollout, and $\Pr(\mathcal{E}_{T,K}) \ge 1 - \zeta$ is a standing condition. The task-bootstrap intervals quantify sampling uncertainty in the empirical summaries; they are distinct from this rollout-event probability.

Quantitative local coupling. Terminal drift correlates with both action discrepancy ($\rho_s = 0.656$, 95% CI $[0.621, 0.691]$) and next-state discrepancy ($\rho_s = 0.657$, $[0.621, 0.692]$). Table 7 shows positive relationships within each system size and both proposal orderings. Across the 18 source-topology groups, the drift–action correlations range from 0.591 to 0.813 and the drift–state correlations from 0.598 to 0.831. Across depth strata, the respective ranges are 0.656–0.669 and 0.656–0.666.

The fixed action–state comparison covers 860 evaluation records: 99.54% coverage with 95% CI $[98.84\%, 100\%]$. Its mean positive slack is 0.0326 ($[0.0272, 0.0380]$), with P95 0.1563. The four exceedances have mean magnitude 0.0873 and maximum 0.1436. Among the 260 records with nonzero action discrepancy, coverage is 98.46% and mean positive slack is 0.1085. The action and state comparisons cover every sampled record in 23 and 22 tasks, respectively; both hold throughout the sampled comparisons of 21 tasks. The composed comparison covers all 864 records, with mean slack 0.3393 and P95 0.7345 (Table 8). Thus its complete empirical coverage comes with greater slack than the action–state comparison. The audit uses all 864 records, including the two at $e_t^a = 1/3$; the displayed action–state panels contain the 862 records at $e_t^a \in \{0, 1/2, 1\}$.

D.3 Figure 4: finite-horizon persistence

Protocol and measured quantities. In ScienceWorld, clean and perturbed continuations share the episode prefix up to a single intervention at $t_0 = 3$. During the first communication pass, an erroneous factual statement is appended to the first solver agent's outgoing message. Statements

Table 7: Terminal drift correlates with both action and next-state discrepancy. Spearman correlations use the complete evaluation records; brackets are 95% confidence intervals from resampling whole tasks. Each row contains 24 tasks.

Group	Records	Drift–action ρ_s	Drift–state ρ_s
All	864	0.656 [0.621, 0.691]	0.657 [0.621, 0.692]
Entry ordering	432	0.661 [0.626, 0.697]	0.663 [0.628, 0.698]
Injection ordering	432	0.650 [0.612, 0.687]	0.651 [0.613, 0.688]
$N = 4$	240	0.680 [0.595, 0.751]	0.683 [0.600, 0.752]
$N = 5$	288	0.669 [0.608, 0.724]	0.669 [0.609, 0.723]
$N = 6$	336	0.633 [0.564, 0.700]	0.632 [0.567, 0.694]

Table 8: Fixed calibration coefficients evaluated on independent tasks. Coverage and its 95% task-bootstrap interval are in percent. Positive slack and exceedance use the known unit metric range ($s_{\mathrm{ref}} = 1$). The first three rows use all 864 records; the last retains the 260 records with nonzero action discrepancy. Mean s includes zero slack; P95 is the empirical 95th percentile.

Comparison	Coverage [CI]	Mean s	P95 s	Exceedances	Max v
Action: $2u$	99.88 [99.65, 100.00]	0.7829	1.6000	1/864	0.1000
State: $\mu_a e^a$	99.54 [98.84, 100.00]	0.0326	0.1563	4/864	0.1436
Composed: $2\mu_a u$	100.00 [100.00, 100.00]	0.3393	0.7345	0/864	0.0000
State, $e^a > 0$	98.46 [96.02, 100.00]	0.1085	0.1641	4/260	0.1436

reverse an observed fact when applicable, with a target-absent fallback otherwise. No further artificial perturbation is injected. We use $N = 5$, one communication pass per task step, and $T = 100$ across seven topologies: chain, tree, layered, leader–follower, bidirectional star, directed ring, and complete graph. Their directed edge counts are 4, 4, 6, 7, 8, 5, and 20, respectively. The analysis retains all 17,934 evaluator-valid task–lag records from 25 to 29 tasks per topology.

The horizon plot measures trajectory divergence through the action-based readout $\delta^{\mathrm{obs}}_{t_0+\ell}$, defined as

$$\delta^{\mathrm{obs}}_{t_0+\ell} = d_{\mathrm{sem}}\left(a^{\mathrm{perturbed}}_{t_0+\ell}, a^{\mathrm{clean}}_{t_0+\ell}\right).$$

Parsed actions use semantic-frame mismatch; other outputs use the recorded deterministic lexical distance. This observable measures paired continuation divergence, while the local experiment measures factual next-state distance. For each task, the horizon average is $97^{-1}\sum_{\ell=1}^{97}\delta^{\mathrm{obs}}_{t_0+\ell}$; the late window is the protocol's final ten lags, 88–97, and the early window is 0–9. We first average within tasks and then across tasks. Late means range from 0.427 to 0.649, and late-minus-early contrasts are positive with 95% intervals above zero for all seven topologies (Table 9). The same direction holds on the 16 tasks common to all seven graphs. Figure 4 retains lag zero separately and groups subsequent lags in bins of five; its bands show 95% task-bootstrap intervals.

Connection to temporal propagation. The common prefix localizes the initial perturbation, while the subsequent continuations reveal its persistence through repeated communication and environment interaction. This is the temporal component of Theorem 1. Its quantitative comparison is obtained by propagating the state envelope from the initial condition and then applying the action-readout interface, as derived in Section D.5. Such a comparison uses coefficients defined for the ScienceWorld reference and action readout. The empirical curves establish persistence in this readout, and the displayed theoretical relation specifies its conditional connection to the state envelope.

D.4 Figure 5: topology–depth propagation factors

Figure 5 evaluates semantic error propagation through controlled factual-error injection during ALFWorld task execution. We insert erroneous factual statements into agent messages at the entrance of the communication block for the transport study, and after each communication update for the accumulation study. These interventions establish known error inputs, enabling us to track how errors are retained and transmitted in subsequent LLM-generated messages. We measure terminal seman-

Table 9: Persistent trajectory deviation in ScienceWorld. Each statistic is averaged within a task and then across tasks. Brackets are 95% task-bootstrap intervals. The horizon average uses lags 1–97; late and early windows use lags 88–97 and 0–9, respectively.

Topology	Tasks	Horizon mean	Late mean	Late − early
Bidirectional star	25	0.479 [0.390, 0.566]	0.608 [0.471, 0.739]	0.531 [0.388, 0.664]
Chain	26	0.458 [0.387, 0.531]	0.638 [0.526, 0.742]	0.594 [0.487, 0.697]
Complete graph	29	0.427 [0.343, 0.511]	0.618 [0.499, 0.730]	0.504 [0.367, 0.632]
Directed ring	26	0.429 [0.339, 0.518]	0.624 [0.496, 0.747]	0.571 [0.433, 0.701]
Layered	25	0.310 [0.223, 0.400]	0.427 [0.301, 0.553]	0.363 [0.225, 0.500]
Leader–follower	27	0.370 [0.278, 0.464]	0.575 [0.443, 0.699]	0.503 [0.361, 0.641]
Tree	25	0.510 [0.416, 0.601]	0.649 [0.535, 0.759]	0.581 [0.470, 0.691]

tic drift from these messages and compare it with the theoretical propagation factors. The operator coefficients are estimated on calibration tasks and held fixed across evaluation configurations.

Protocol and analytical reference. We evaluate 18 strongly connected directed graphs: four with $(N, E) = (4, 5)$, ten with $(5, 8)$, and four with $(6, 12)$. A communication configuration is one graph–depth pair $(\mathcal{G}, K)$; N and E are properties of $\mathcal{G}$. Figure 5 uses $K = 2, 3, 4$ for $N = 4, 6$ and $K = 2, 3, 4, 5$ for $N = 5$, giving $4 \times 3 + 10 \times 4 + 4 \times 3 = 64$ configurations per channel. Each configuration uses the same 12 ALFWorld tasks, two per task family, so each panel summarizes 768 task observations. The accompanying graph registry specifies the directed edges and the graph–depth combinations used in each study. Observed terminal drift is the sum of the values parsed from the LLM messages.

We inject erroneous factual statements into agent messages during task execution. Injection at the entrance of the communication block examines initial-error transport, while injection after each communication update examines error accumulation. Observed terminal drift is obtained by summing the semantic-drift measurements across agents' terminal messages. The horizontal axis shows the theoretical value computed using the fixed calibrated operator, and the vertical axis shows terminal semantic drift measured from the generated natural-language messages. The diagonal $y = x$ indicates numerical agreement between these quantities. The lower panels display $100r_i$, where $r_i = (y_i - x_i)/x_i$ for the fixed positive comparison value x_i and observed condition mean y_i.

Prediction ordering and numerical scale. The transport and accumulation factors achieve Spearman correlations 0.9982 and 0.9988, respectively. Their median absolute relative deviations are 0.56% and 0.00%, with P95 values 11.60% and 11.32%; the median signed deviation is zero in both channels (Table 10). Correlation intervals resample tasks within their families, retaining every configuration of a sampled task and recomputing the condition means. Across fixed-depth strata, transport correlations range from 0.936 to 0.991 and accumulation correlations from 0.891 to 1.000. Matched graph contrasts at fixed N, E, K are positive in six of six transport and accumulation pairs. These results support the numerical scale and align the interpretation of Figure Figure 5 with its intervention structural dependence of the two propagation factors under factual-error injection into natural-language communication. The complete experimental code is available in our anonymous repository. It includes the procedure for computing semantic drift from generated natural-language messages, the parsing rules for extracting numerical drift values, and the calibration procedure for the communication coefficients (d) and (g). These implementations support reproduction of the measurements and results reported in Figure 5.

D.5 Cross-figure interpretation

Connection to the closed-loop law. The communication and local-interface measurements address consecutive parts of the derivation. The horizon observable connects to the state bound through the action interface. With the representation and reference fixed for that comparison, the theorem's standing conditions give

$$e_t^a \le L_{\mathcal{G},K}^a \delta_t + r_{\mathcal{G},K}^a,$$

Table 10: Agreement with the fixed analytical communication responses. Relative deviations are $r = (y - x)/x$ and are reported in percent; x is held fixed. Correlation intervals resample tasks within their families and recompute condition means.

Channel / size	Conditions	ρ_s [95% CI]	Med. $\lvert r\rvert$	P95 $\lvert r\rvert$	Med. r
Transport	64	0.9982 [0.9981, 0.9984]	0.56	11.60	0.00
$N = 4$	12	0.9803 [0.9803, 0.9875]	0.00	4.69	0.00
$N = 5$	40	0.9971 [0.9968, 0.9973]	2.06	12.01	0.00
$N = 6$	12	1.0000 [1.0000, 1.0000]	0.65	7.43	-0.65
Accumulation	64	0.9988 [0.9987, 0.9988]	0.00	11.32	0.00
$N = 4$	12	1.0000 [1.0000, 1.0000]	0.00	0.00	0.00
$N = 5$	40	0.9959 [0.9959, 0.9960]	0.00	12.99	0.00
$N = 6$	12	1.0000 [1.0000, 1.0000]	2.95	6.56	-2.95

Table 11: Configuration and sample counts for the two studies. N is the solver count and E is the directed edge count. Distinct graphs counts adjacency structures at fixed (N, E); pairs counts the graph–depth combinations actually evaluated. The local $N = 5$ sample contains 24 selected pairs within $K = 1, 2, 3$. Local decision records include both orderings and all $N + 1$ doses: $24 \times 2 \times (N + 1)$ per size.

			Figure 5: communication		Figure 3: local interfaces		
N	E	Distinct graphs	K	Graph–depth pairs	K	Source pairs	Decision records
4	5	4	2–4	12	2–4	12	240
5	8	10	2–5	40	1–3	24	288
6	12	4	2–4	12	2–4	12	336
Total: 18 graphs				64		48	864

where

$$L^a_{\mathcal{G},K} = \lambda_s + \lambda_c\big(c_0 Q_{\mathcal{G}}(K) + b S_{\mathcal{G}}(K)\big),$$
$$r^a_{\mathcal{G},K} = \lambda_c\big(\bar{\xi} Q_{\mathcal{G}}(K) + \bar{\varepsilon} S_{\mathcal{G}}(K)\big) + \bar{\nu}.$$

We use the compact coefficients of the main-text theorem:

$$a_{\mathcal{G},K} = \mu_0 + \mu_1\big(Q_{\mathcal{G}}(K) + S_{\mathcal{G}}(K)\big),$$
$$\gamma_{\mathcal{G},K} = \varepsilon_0 + \varepsilon_1\big(Q_{\mathcal{G}}(K) + S_{\mathcal{G}}(K)\big),$$

where $\mu_0 = \mu_s + \mu_a\lambda_s$, $\mu_1 = (c_0 + b)\mu_a\lambda_c$, $\varepsilon_0 = \mu_a\bar{\nu} + \bar{\eta}$, and $\varepsilon_1 = (\bar{\xi} + \bar{\varepsilon})\mu_a\lambda_c$. The separate-factor action coefficients L^a, r^a and the compact state coefficients a, γ therefore keep their respective meanings. Substitution of Theorem 1, with the recursion started at t_0, gives on the same rollout event

$$e^a_{t_0+\ell} \le L^a_{\mathcal{G},K}\left(a^{\ell}_{\mathcal{G},K}\delta_{t_0} + \gamma_{\mathcal{G},K}\sum_{j=0}^{\ell-1} a^j_{\mathcal{G},K}\right) + r^a_{\mathcal{G},K}.$$

When the action interface is stated for the measured paired cost, $e^a_t = e^{\mathrm{obs}}_t$ gives the corresponding action-readout comparison. The deterministic action-law interpretation above provides one such instance. For random execution, this connection requires the paired-cost interface on the stated rollout event. Each experiment uses its specified representation and reference, with coefficients defined for that instance. The empirical results assess propagation-factor scaling under controlled factual-error injection, quantify local interface coverage and slack on separate tasks, and show persistent deviation during finite-horizon interaction.

D.6 Shared implementation and statistical protocol

Inference and shared parameters. Models are served locally through vLLM. The communication and local decision experiments use temperature zero, request seed 20260905, and output limits

Table 12: Parameters shared within each experimental protocol.

Quantity	Value	Scope and source
$\widehat{d}$ and $\widehat{g}$	$0.83258, 0.90538$	All 64 communication configurations
u_0, u_{inj}	$1, 1$	Unit injections in the two communication channels
λ_c	2	All local evaluations; six calibration tasks
μ_a	0.3922413793	All local evaluations; six calibration tasks
$\bar{\nu}, \bar{\eta}$	$0, 0$	Through-origin local comparisons
s_{ref}	1	Known range of the action and state metrics

of 192 and 48 tokens, respectively. Their shared parameters are listed in Table 12. The supplementary ALFWorld horizon experiment uses a single observation perturbation at $t_0 = 3$, complete graphs with $N = 2, \ldots, 8$, $K = 5$, and $T = 49$; its task-level analysis is kept separate from ScienceWorld.

E Experiment

E.1 Experiment setup and Benchmark

Benchmark Descriptions. We evaluate our method on five benchmarks spanning four complementary task regimes: household interaction, interactive scientific reasoning, knowledge-intensive reasoning, and office automation. Together, these benchmarks cover partially observable environments, sparse and delayed feedback, multi-source evidence synthesis, and cross-application execution. For each benchmark, we follow the evaluation settings used in the corresponding baseline studies (Zhang et al., 2025a; Kang et al., 2026).

- **ALFWorld** (Shridhar et al., 2021) (available at `https://github.com/alfworld/alfworld`) translates visually grounded household tasks from ALFRED into interactive text environments. An agent must navigate a partially observed home, locate and manipulate objects, and execute temporally ordered operations such as cleaning, heating, cooling, and placement. Success therefore requires grounded planning and persistent tracking of object locations, states, and previously executed actions across long interaction trajectories.
- **ScienceWorld** (Wang et al., 2022) (available at `https://github.com/allenai/ScienceWorld`) is an interactive text environment built around an elementary-school science curriculum. Its task families span topics such as phase transitions, electrical circuits, chemistry, biology, and mechanics. Rather than answering science questions directly, agents must demonstrate procedural understanding by navigating the environment, assembling experimental setups, manipulating materials, and interpreting intermediate outcomes. ScienceWorld thus evaluates whether an agent can retain and operationalize scientific knowledge over extended action sequences.
- **FEVER** (Thorne et al., 2018) (available at `https://fever.ai/dataset/fever.html`, distributed under the applicable Wikipedia) contains 185,445 claims annotated as Supports, Refutes, or NotEnoughInfo, together with sentence-level evidence for verifiable claims. In our interactive formulation, agents retrieve and inspect Wikipedia evidence before committing to a verdict. The benchmark separates successful reasoning into evidence acquisition, cross-sentence aggregation, and recognition of cases for which the available evidence is insufficient.
- **HotpotQA** (Yang et al., 2018) (available at `https://hotpotqa.github.io/`) comprises approximately 113K natural questions designed to require reasoning across multiple Wikipedia documents. It includes both bridge questions, which connect entities through intermediate evidence, and comparison questions, which require aligning facts about multiple entities. Sentence-level supporting-fact annotations make the benchmark suitable for evaluating not only answer correctness, but also whether an agent retrieves and preserves the evidence needed for an explainable multi-hop inference.
- **OfficeBench** (Wang et al., 2024) (available at `https://github.com/zlwang-cs/OfficeBench`) evaluates language agents on realistic office workflows involving applications such as Word, Excel, PDF, Email, and Calendar. Tasks require agents to plan multi-step workflows, transfer information across applications, and ground actions in a large heterogeneous tool

space. Outputs are assessed through task-specific exact-match, fuzzy-match, and execution-based checks, making success contingent on the final artifacts and environment state rather than on the agent's textual declaration of completion.

These benchmarks expose complementary memory demands. ALFWorld and ScienceWorld require agents to preserve evolving environment state. FEVER and HotpotQA require evidence to remain available across retrieval and reasoning steps. and OfficeBench requires intermediate information and execution state to remain coherent across heterogeneous tools.

Evaluation Metrics. We report task success rate for ALFWorld, mean final subgoal completion for ScienceWorld, exact label accuracy for FEVER, answer exact match for HotpotQA, and task pass rate for OfficeBench. An OfficeBench task is successful only when all task-specific checks on the resulting artifacts and environment state are satisfied.

Implementation details. All experiments are conducted on a server equipped with eight NVIDIA H100 GPUs, each with 80 GB of memory. Our implementation uses Python 3.12 and PyTorch with CUDA 12.8. We serve the Qwen3 models locally through vLLM in bfloat16 precision, with each model instance running on a single GPU and independent experimental runs distributed across the available GPUs. The model sizes, decoding settings, and task-specific configurations are specified in the corresponding experimental descriptions. We follow the evaluation protocols established in prior work (Zhang et al., 2025a) and reproduce the baselines according to the experimental settings reported in their original papers. The detailed implementation and experiment configurations are provided at `https://anonymous.4open.science/r/RECOVER-8888/README.md`.

E.2 Evaluating the bridge from proposed theory to RECOVER

Our experiments evaluate the practical consequences of the theoretical design criteria. Theorems 2 and 3 motivate preserving earlier evidence when endogenous interaction cannot recover information lost from the transient input. The decision-identifiability proposition and the sufficient conditions in Appendix B.6 guide **RECOVER** to retain evidence for precondition queries and inject it when that evidence resolves an undetermined verdict. We evaluate these choices through task performance, comparisons with fixed injection schedules, and token and storage costs. **RECOVER** implements these criteria as an integrated memory procedure. The experiments in this paper evaluate its task performance, injection strategy, and efficiency. And the five worked examples stated in Appendix C apply the shared evidence procedure to recorded source observations under specified recipient views. They illustrate fact extraction, validity updates, and insertion decisions.

Table 1 reports the highest mean performance for **RECOVER** across all 15 benchmark–framework combinations. Relative gains over the strongest memory baseline range from 1.45% to 7.31%; the largest gain over raw history is 33.02% on FEVER with DyLAN. RECOVER uses only in-task memory. Across the settings in Table 1 and Figure 7, some memory methods underperform raw history. On ALFWorld with AutoGen, raw history achieves 75.597%, compared with 68.657% for HiAgent and 91.542% for RECOVER. These comparisons assess complete memory procedures. The results support the end-to-end effectiveness of RECOVER's theory-guided design. The injection-schedule (Figure 6) comparison supports the practical value of the sufficiency-based trigger: supplying retained evidence when it resolves a current query achieves a better accuracy–token trade-off than the evaluated fixed schedules.

This paper's analysis provides testable constraints on memory representation and intervention. A quantitative mapping from propagation magnitude to task-level recovery gain has shown by the existing results. And existing results already distinguish concrete algorithmic choices. First, Proposition 1 specifies which distinctions a memory representation must preserve. If two histories yield the same transient input but require different verdicts, a memory representation that merges them cannot support exact recovery on both. This gives a concrete criterion for what to remember: preserve decision-relevant distinctions that the transient input leaves unresolved. RECOVER operationalizes this criterion by retaining evidence for candidate-action preconditions and updating its validity as the task state changes. The criterion can be tested using paired histories with identical current inputs and different required verdicts. Second, the analysis leads to a selective rule for when to inject. RECOVER injects evidence when the current precondition check is undetermined and the retained evidence resolves it. This distinguishes sufficient evidence from merely relevant content, and evidence-

Table 13: Performance results with Qwen3-8B.

MAS	Memory Cross-task In-task	ALFWorld	FEVER	SciWorld	Office	HotpotQA
AutoGen	Raw history	8.955%	14.179%	22.384%	9.783%	7.887%
	GMemory	40.299%	21.642%	13.818%	14.607%	13.059%
	MetaGPT	26.119%	24.627%	28.691%	18.391%	14.112%
	Voyager	17.910%	20.149%	27.083%	15.909%	12.032%
	Generative	29.851%	23.134%	28.469%	17.978%	14.180%
	MemoryBank	14.179%	11.194%	11.080%	7.792%	6.226%
	ChatDev	26.866%	22.388%	27.132%	17.391%	13.437%
	ACON	23.881%	14.179%	11.111%	8.511%	6.941%
	HiAgent	15.672%	8.955%	13.333%	9.211%	6.982%
	Ours	67.164%	39.552%	30.537%	28.736%	23.606%
DyLAN	Raw history	5.970%	12.687%	19.794%	10.870%	8.400%
	GMemory	43.284%	18.657%	7.385%	14.773%	11.060%
	MetaGPT	14.925%	21.642%	26.179%	17.391%	14.841%
	Voyager	15.672%	20.896%	23.091%	14.607%	13.113%
	Generative	15.672%	21.642%	28.335%	18.478%	14.868%
	MemoryBank	14.179%	15.672%	11.072%	11.688%	9.804%
	ChatDev	8.955%	17.164%	22.280%	11.828%	10.169%
	ACON	22.388%	17.164%	13.333%	11.702%	9.088%
	HiAgent	9.702%	10.448%	11.111%	8.955%	6.779%
	Ours	64.179%	39.552%	29.908%	30.667%	25.564%
MacNet	Raw history	5.970%	14.925%	23.525%	8.696%	9.372%
	GMemory	34.328%	17.910%	12.489%	13.483%	12.883%
	MetaGPT	11.940%	18.657%	32.593%	14.130%	12.100%
	Voyager	13.433%	20.149%	25.781%	13.483%	13.342%
	Generative	19.403%	21.642%	33.326%	15.730%	15.571%
	MemoryBank	11.940%	10.448%	17.572%	9.091%	8.548%
	ChatDev	13.433%	17.910%	27.580%	12.903%	11.992%
	ACON	19.403%	11.194%	13.673%	8.511%	8.845%
	HiAgent	11.940%	10.448%	13.689%	7.463%	7.184%
	Ours	55.224%	37.313%	33.982%	25.333%	24.713%

dependent intervention from fixed-frequency retrieval. Appendix B.6 establishes correctness of the resulting local verdict under explicit evidence-validity, recognition, and retrieval conditions. These choices also have observable empirical consequences. The timing experiments compare the resulting insertion policy with fixed schedules and report improved accuracy–token trade-offs. They evaluate a concrete algorithmic consequence of the design criterion, while the benchmark experiments assess the integrated method. The theoretical contribution therefore includes a recoverability boundary, a characterization of sufficient memory, and an explicit criterion for evidence-based intervention. Theorems proposed in this paper characterizes which decision-relevant distinctions memory must preserve. RECOVER translates this criterion into evidence retention and selective injection, whose task-level benefits are evaluated empirically.

E.3 Additional Experiment Results

Robustness across backbone scales. We further evaluate our method with Qwen3-8B and Qwen3-32B to assess its robustness to changes in backbone capacity. As shown in Tables 13 and 14, our method achieves the highest performance across all five benchmarks under AutoGen, DyLAN, and MacNet with both backbones. Together with the Qwen3-14B results, these findings demonstrate consistent effectiveness across the model sizes, with improvements maintained for both smaller and larger backbones.

Table 14: Performance results with Qwen3-32B.

MAS	Memory (Cross-task / In-task)	ALFWorld	FEVER	SciWorld	Office	HotpotQA
AutoGen	Raw history	88.060%	62.687%	57.192%	43.699%	34.922%
	GMemory	88.060%	64.179%	73.109%	46.221%	40.797%
	MetaGPT	87.313%	70.149%	72.861%	52.400%	40.459%
	Voyager	90.776%	67.910%	72.815%	53.690%	41.472%
	Generative	89.791%	67.910%	73.933%	50.454%	40.014%
	MemoryBank	79.851%	65.672%	59.915%	44.375%	35.436%
	ChatDev	88.806%	58.955%	54.055%	45.834%	34.787%
	ACON	90.537%	66.418%	16.224%	30.268%	24.551%
	HiAgent	70.134%	45.522%	15.229%	33.952%	25.307%
	Ours	92.537%	71.642%	75.434%	55.161%	42.255%
DyLAN	Raw history	85.821%	70.896%	52.409%	52.143%	40.513%
	GMemory	90.299%	72.149%	71.458%	57.103%	41.553%
	MetaGPT	89.552%	65.672%	70.664%	57.194%	45.615%
	Voyager	91.776%	64.179%	70.514%	54.162%	47.049%
	Generative	90.299%	70.896%	70.849%	57.426%	46.521%
	MemoryBank	79.851%	70.896%	56.837%	56.688%	44.683%
	ChatDev	83.582%	70.149%	54.639%	47.775%	40.365%
	ACON	90.299%	73.881%	34.071%	41.949%	33.369%
	HiAgent	70.896%	61.940%	19.168%	37.520%	30.466%
	Ours	94.403%	75.373%	76.647%	59.180%	49.426%
MacNet	Raw history	70.156%	67.910%	67.967%	41.593%	43.444%
	GMemory	82.273%	61.940%	76.758%	46.523%	44.186%
	MetaGPT	58.521%	58.955%	74.858%	43.531%	37.556%
	Voyager	82.309%	57.149%	65.716%	46.121%	43.968%
	Generative	78.459%	66.418%	77.204%	45.999%	44.846%
	MemoryBank	82.327%	56.716%	78.838%	46.099%	45.469%
	ChatDev	63.239%	56.716%	66.027%	40.564%	31.924%
	ACON	59.546%	39.552%	59.466%	29.266%	37.880%
	HiAgent	53.000%	41.791%	49.740%	27.984%	29.048%
	Ours	85.791%	69.403%	79.466%	48.454%	46.063%

Performance across tasks and multi-agent frameworks. We evaluate memory methods on ALFWorld, ScienceWorld, and FEVER using Qwen3-14B with AutoGen and DyLAN. Our method achieves the highest performance in all six settings, outperforming the strongest available baseline by 1.67–5.22 percentage points and raw history by 8.79–16.01 points. On ALFWorld (Figures 11 and 12), it achieves success rates of 91.54% with AutoGen and 92.54% with DyLAN. On ScienceWorld (Figures 15 and 16), the mean subgoal completion scores reach 72.90% and 56.81%, respectively, exceeding ACON, the strongest baseline in both settings, by 4.97 and 3.61 points. On FEVER (Figures 13 and 14), our method attains accuracies of 68.33% and 64.49%, respectively. These results show consistent performance gains across both frameworks and tasks with different evaluation criteria.

Token and storage efficiency. The ALFWorld results further show that these performance gains are accompanied by lower token consumption (Figures 11 and 12). Our method uses the fewest total tokens among the evaluated methods: 38.4k per task with AutoGen and 82.9k with DyLAN, reducing consumption relative to GMemory by 55.6% and 50.2%, respectively. Peak serialized memory payload also drops from 32.84 to 1.02 MiB with AutoGen and from 65.40 to 2.06 MiB with DyLAN, a reduction of approximately 97% in both settings. Compared with raw history, our method reduces token consumption by 27.3% and 18.4%, respectively. Thus, on ALFWorld, our method combines higher task performance and lower token consumption with a small additional memory footprint relative to raw history.

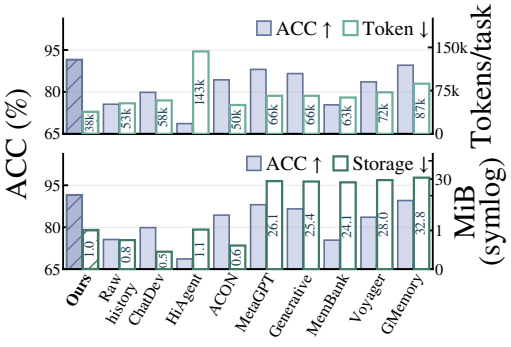

Figure 11: ALFWorld performance with AutoGen using Qwen3-14B, paired with token usage (top) and memory storage (bottom).

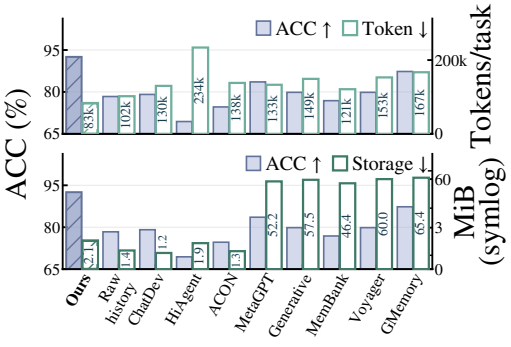

Figure 12: ALFWorld performance with DyLAN using Qwen3-14B, paired with token usage (top) and memory storage (bottom).

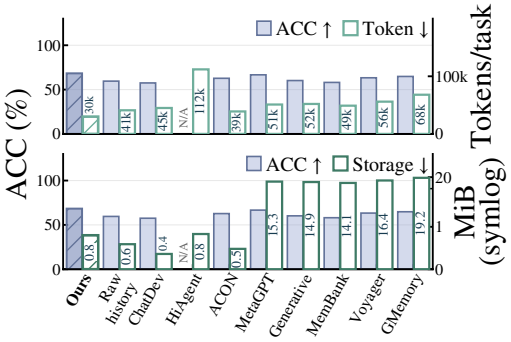

Figure 13: FEVER performance with AutoGen using Qwen3-14B, paired with token usage (top) and memory storage (bottom).

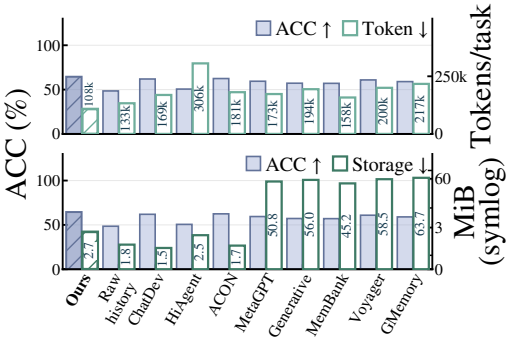

Figure 14: FEVER performance with DyLAN using Qwen3-14B, paired with token usage (top) and memory storage (bottom).

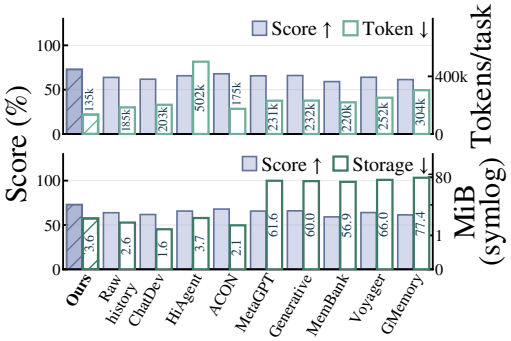

Figure 15: ScienceWorld performance with AutoGen using Qwen3-14B, paired with token usage (top) and memory storage (bottom).

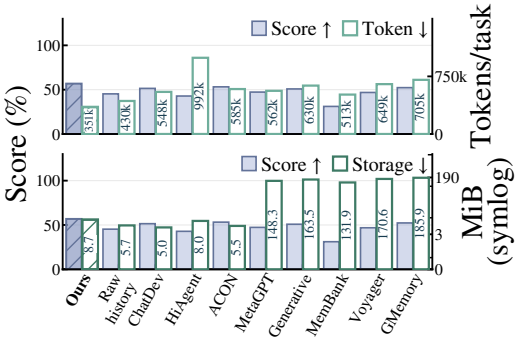

Figure 16: ScienceWorld performance with DyLAN using Qwen3-14B, paired with token usage (top) and memory storage (bottom).