WHEN HANDOFFS GO STALE: VALIDITY-AWARE STATE TRANSFER FOR MULTI-AGENT LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent LLM systems rely on handoffs to continue tasks across specialized roles. To fit limited context budgets, existing methods summarize or retrieve task history, but may omit constraints, verification status, and dependencies even when relevant facts are retained. We formulate this gap as a receiver-conditioned state-projection problem and propose a training-free method for validity-aware state transfer. The method maintains a typed task-state graph that tracks provenance, evidence, and lifecycle changes, constructs budgeted capsules containing action-relevant state and its validity relations, and repairs affected state slots using execution feedback. Experiments on SWE-bench Lite show that our method achieves a 63.3% resolution rate, compared with 40.7% for vector memory, the strongest evaluated multi-agent baseline by task success, while reducing total tokens per resolved instance by 46.6%. Further controlled evaluations across software repair, knowledge analysis, and enterprise workflows demonstrate improved retention of critical state, identification of stale information, and recovery from handoff errors relative to summarization and retrieval baselines.

1 INTRODUCTION

Large language model systems are increasingly used as teams of specialized agents for complex tasks such as software engineering, code debugging, and data analysis (Wu et al., 2023; Qian et al., 2024). These systems divide a task among agents with different instructions, tools, or expertise. When the current agent reaches a stage that is better handled by another agent, it can transfer control together with the context needed to continue; this transfer is commonly called a *handoff* (Barrak, 2025; KC & Budathoki, 2026). Handoff allows a multi-agent system to continue a task from an intermediate state without requiring the receiving agent to reconstruct the entire preceding interaction, making it a basic mechanism for multi-agent task execution and interrupted-task continuation.

Current handoff systems pass raw trajectories or selected messages, compress them into summaries, or encode intermediate results in structured notes and typed representations. Persistent memory adds provenance, conflict handling, validity tracking, and validated writes across sessions (Zhou et al., 2026; Volkov et al., 2026; Li et al., 2026b). Under limited budgets, these interfaces balance completeness, cost, and usability, but topical relevance alone cannot ensure that the receiver retains the constraints, verification status, and dependencies needed to act. Recent studies report costly rediscovery even in shared repositories (KC & Budathoki, 2026), and receiver-dependent benefits from inherited trajectories (Ganz et al., 2026). Controlled studies further show that compression can weaken blockers into caveats (Sun et al., 2026), loss of an "unverified" status can increase risky-action approval (Hu, 2026), and use boundaries can disappear while facts survive (Wang et al., 2026). Engineering experience with long-running coding agents likewise highlights repeated rediscovery and premature completion when progress records and testing procedures are missing (Anthropic, 2025). These findings motivate selecting a receiver-specific view of evolving task state that preserves both relevant information and the conditions needed for its safe use.

A simple approval workflow illustrates the gap (Figure 1). A reviewer records "approval pending" and requires the next agent to await approval and remain read-only. If compression drops these requirements, the receiver may treat the surviving fact as a reminder and perform a forbidden write. We call this loss of action-relevant conditions an *invalid handoff*; *stale state* is the specific case in which time, version, or scope changes make earlier information inapplicable. This distinguishes

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.

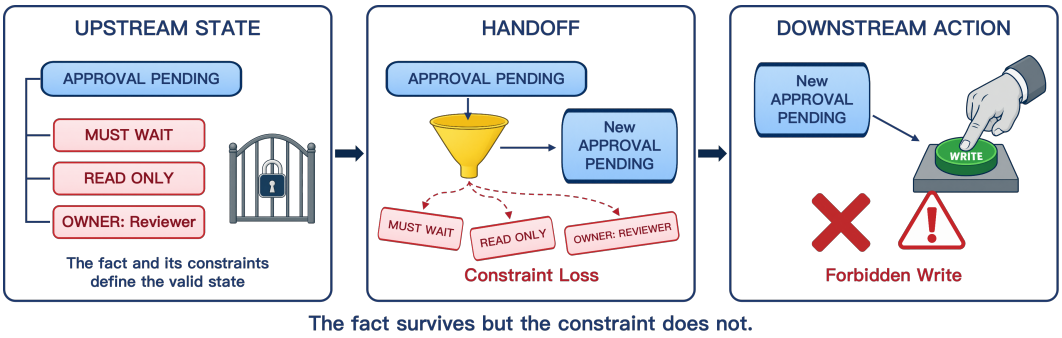

Figure 1: **A handoff can preserve a fact while losing its action-binding constraints.** The upstream state contains "approval pending" together with requirements to wait, remain read-only, and defer to the reviewer. The handoff preserves the fact but drops these constraints, after which the successor performs a forbidden write.

remembering state from *validly handing off state*: persistent memory may retain facts and updates, while the receiver's view omits the evidence, dependencies, or use boundaries needed to interpret them. We therefore formulate budgeted handoff as receiver-conditioned state projection: given an evolving task state, a receiving role, and a communication budget, construct a bounded capsule that preserves the action-relevant state and the validity relations needed to support the receiver's next action.

We propose *validity-aware state transfer* to preserve these conditions across handoffs. A typed state graph records the scope, evidence, and lifecycle of each item. Receiver-conditioned selection treats the qualifications needed to interpret an item as part of its communication cost. When execution feedback exposes an information gap, provenance links direct repair to the affected slot and its source. This design makes applicability explicit at the point where task state becomes input to another agent.

Our experiments combine software repair on SWE-bench Lite (Jimenez et al., 2024) with controlled handoff tasks spanning software repair, knowledge analysis, and approval workflows. On SWE-bench Lite, validity-aware transfer achieves a 63.33% resolution rate, compared with 40.67% for vector memory, the strongest evaluated multi-agent baseline by task success, while reducing total tokens per resolved instance by 46.6%. On controlled handoff failures, it achieves 56.47% repair success versus 29.11% for the same baseline, with fewer additional rounds and rework events. State-level analyses also show better identification of stale information and more accurate preservation of support, conflict, and supersession relations than summary and retrieval baselines.

Our contributions are as follows:

- We present a formulation of budgeted handoff as receiver-conditioned state projection. This formulation separates the information a projection retains from the validity conditions the receiver's next action requires, and thereby helps to explain when a handoff remains actionable and when it does not.
- We develop validity-aware state transfer, which integrates typed state graphs, lifecycle tracking, receiver-conditioned capsule selection, and feedback-guided slot-level repair. Keeping the scope, evidence, and status of an item attached to the item itself allows each receiver to separate the state it may still act on from the state that has gone stale, and routes feedback to the specific slots that need repair.
- We show that validity-aware state transfer achieves higher task success and more effective handoff recovery than the evaluated baselines, while reducing the total token cost per resolved instance. This indicates that conditioning transfer on the receiver's validity conditions can improve effectiveness and efficiency at the same time.

2 PRELIMINARIES: HANDOFF-BASED COLLABORATION

2.1 COLLABORATION THROUGH HANDOFFS

We consider a complex task g executed by agents $\mathcal{A} = \{a_1, \ldots, a_n\}$ with specialized roles, instructions, and tools. Each agent works on an assigned step and produces observations, intermediate results, or changes to shared artifacts. A *handoff* transfers control over the next step to a receiving agent together with the context needed to continue (Liu et al., 2026; Rui et al., 2026). The receiver may perform several reasoning and tool-use steps before handing off again, and later feedback may return the task to an earlier role. We study how state is composed and managed across these transitions, taking the workflow's role assignments and choice of receiver as given. Let H_t denote the task history available at handoff boundary t, including messages, tool results, and verification feedback.

2.2 HANDOFF CONTEXT AND BUDGET

Let $\mathcal{S}_t$ denote the task state maintained from H_t, which may be organized as a trajectory, a summary, or a structured store. At boundary t, the workflow specifies receiver a and context x_t, comprising the task stage, assigned next step, and observed artifact versions. The receiver also has existing context $K_t(a)$, such as its task and role instructions and prior observations. The transferred content is a *handoff capsule* h_{t+1}, constructed by a state-transfer mechanism Γ under token budget B_{t+1}:

$$h_{t+1} = \Gamma(\mathcal{S}_t, x_t, a, K_t(a), B_{t+1}), \qquad \mathrm{cost}(h_{t+1}) \le B_{t+1}. \tag{1}$$

The receiver acts using the combined context $C_t(h, a) = (K_t(a), h)$. Cost counts the complete serialized capsule with the receiver's tokenizer, including any status annotations, evidence pointers, and relation statements it carries. The capsule budget limits what is transferred at this boundary; subsequent reasoning, tool calls, and additional requests also contribute to total execution cost. Access to a shared repository or document does not place its contents in $K_t(a)$: the receiver must read the relevant material if it is needed to interpret the handoff.

2.3 TASK-LEVEL OBJECTIVE

Our objective is to improve the reliable completion of complex collaborative tasks through effective handoffs under limited communication budgets. Let $\mathcal{M}$ denote the handoff approach evaluated within the workflow of Section 2.1, with agent roles, models, and tool access held fixed. For task g, let $Y_g^{\mathcal{M}} \in \{0, 1\}$ indicate whether the final outcome satisfies the task's success criterion, and let $N_g^{\mathcal{M}}$ count all input and output tokens consumed during the run. We use their expectations to measure task success and mean total token cost across the evaluated tasks:

$$J(\mathcal{M}) = \mathbb{E}\left[Y_g^{\mathcal{M}}\right], \qquad \mathcal{C}(\mathcal{M}) = \mathbb{E}\left[N_g^{\mathcal{M}}\right]. \tag{2}$$

Expectations are over the task distribution and execution randomness. We seek higher J under the per-handoff budgets in Equation (1) and common workflow execution limits, and report $\mathcal{C}$ to characterize the tradeoff between success and cost. Token accounting covers all model calls required by a run, including context preparation, agent execution, retries, and auxiliary requests. Handoff effectiveness is thus assessed through its contribution to the final task outcome and the resources consumed along the way. State-level diagnostics help explain information loss and recovery within this process; Appendix B specifies the corresponding empirical measures.

3 VALIDITY-AWARE STATE TRANSFER

3.1 OVERALL FRAMEWORK

During execution, new observations may change whether earlier state remains usable. We therefore update state before constructing a capsule and repair gaps exposed by feedback. Figure 2 links these steps:

$$\mathcal{M} = (U, \Gamma, \mathcal{R}), \qquad \mathcal{S}_{t+1} = U(\mathcal{S}_t, \Delta H_{t+1}). \tag{3}$$

Here, U incrementally incorporates ΔH_{t+1}, the observations between boundaries t and $t+1$, including execution feedback and repair responses; Γ constructs capsules, and $\mathcal{R}$ handles repair. The method requires no training and uses the workflow's roles, execution order, and tools. The following subsections address three questions:

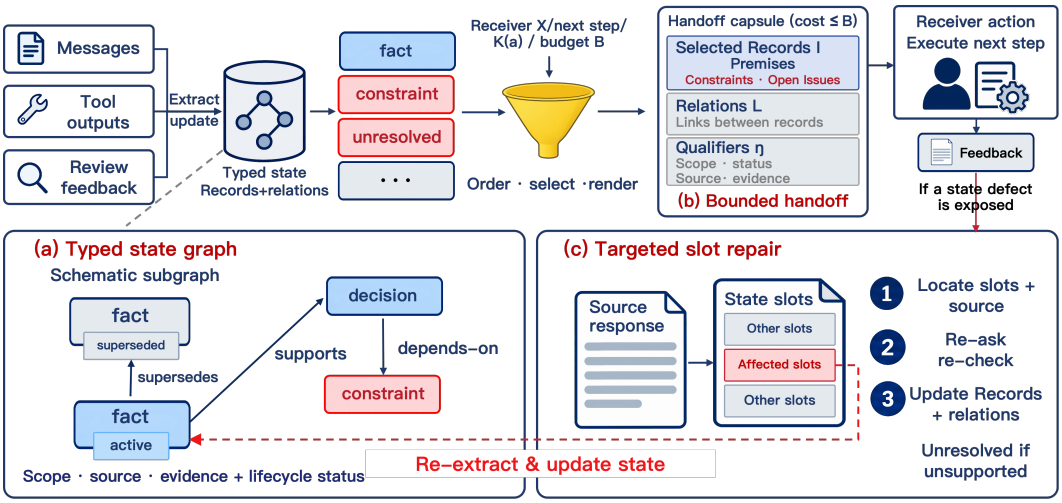

Figure 2: **Validity-aware state transfer.** (a) Messages, tool outputs, and review feedback update a typed state graph with lifecycle status and support, dependency, and supersession relations. (b) Receiver-conditioned selection constructs a capsule within budget B, retaining selected records, their relations, and scope, status, source, and evidence qualifiers. (c) Execution feedback guides targeted repair of affected slots through source identification, re-asking or re-checking, and updates to records and relations. Unsupported items remain unresolved.

1. How can the current state be used? (Section 3.2)
2. What should be passed to the receiver under a limited budget? (Section 3.3)
3. What is missing, which source should be queried, and when should repair stop? (Section 3.4)

3.2 State Representation and Update Rules

The task state is a typed graph $\mathcal{S}_t = (E_t, R_t)$, with records E_t and relations R_t. Each record forms

$$e = \langle k, v, \tau, \mathrm{src}, \mathrm{ev}, \omega, t_e, \delta \rangle, \tag{4}$$

where k is a semantic slot, v its content, and τ its type:

$$\tau \in \{\texttt{fact}, \texttt{constraint}, \texttt{hypothesis}, \texttt{decision}, \texttt{tool_state_delta}, \texttt{artifact}, \texttt{unresolved_issue}\}. \tag{5}$$

The source src identifies the producing agent or event; ev points to supporting messages, logs, or artifacts and is empty when evidence is unavailable. Scope ω specifies the artifact version, task stage, and permitted uses. The creation round t_e and optional expiry condition δ capture temporal applicability. A slot can contain several records with different scopes: a passing test on v_1 and a pending check on v_2 describe distinct states.

Extraction. An off-the-shelf LLM extracts candidate records and relations from free-form messages, conditioned on the task goal, producing role, and existing state; rules parse explicit tool outcomes and structured fields. The extractor preserves the distinction between a hypothesis, a reported decision, and an observed result, and attaches evidence pointers to their sources. Candidates must pass a schema check before entering the graph. Malformed output is returned for correction, with rule extraction as a fallback if parsing still fails.

Reconciliation. Operator U compares records by slot, content, provenance, and scope. Equivalent content within the same scope is consolidated with its evidence. A tool result, verification feedback, or explicit revision that establishes a replacement for the same slot and intended use creates a `supersedes` edge from the new record to the old one. Incompatible claims under overlapping scopes receive a `contradicts` edge and retain the conflict until evidence supports a resolution; recency alone does not establish which claim is correct. A `supports` edge points from evidence to a claim, and a `depends-on` edge from a claim or decision to its prerequisite. A result for a new version does not overwrite the evidence recorded for an earlier version.

Lifecycle and applicability. The maintained status $\sigma_t(e)$ is ACTIVE for a current record, SUPERSEDED after replacement, CONTRADICTED when incompatible evidence appears, EXPIRED when δ holds, or UNRESOLVED while a check or decision is pending. Expiry removes current applicability without proving the original content false. Status also does not replace evidence: an ACTIVE hypothesis remains a hypothesis. For a claim to be presented as established grounds for the next action, its scope must match that use, its required support and prerequisites must hold, and no unresolved conflict, replacement, or expiry may invalidate that use. Historical and open records can still be conveyed with explicit qualifications. The graph retains their evidence and relations even when they are excluded from a capsule.

3.3 RECEIVER-CONDITIONED CAPSULE CONSTRUCTION

The constructor Γ takes $\mathcal{S}_t$, next-step context x_t, receiver a, existing context $K_t(a)$, and budget $B = B_{t+1}$. It considers records visible to a and relevant to the assigned step, including constraints and unresolved blockers. Executors emphasize decisions, artifacts, and tool state; verifiers emphasize checking conditions and evidence. Applicability determines how a record may be used, while the following score determines its selection priority.

Priority score. Task-template weights w define a priority per estimated token:

$$\rho_t(e, a) = \frac{w^\top \phi_t(e, a)}{\widehat{c}(e)}, \qquad \widehat{c}(e) > 0. \tag{6}$$

The normalized features ϕ_t encode role relevance, task priority, recency, assessed confidence and status, evidence availability, verification or tool signals, and negative estimated token cost. Role relevance matches record types to receiver roles; priority emphasizes constraints and open issues; temporal features use creation time, expiry, and replacement information. The denominator $\widehat{c}(e)$ estimates rendered record length.

Packing records with their qualifications. Candidates are scanned in descending ρ_t order. For each candidate, the renderer includes its type, applicable scope and status, and the evidence or dependency relations needed for its stated use. A claim cannot retain an established status while losing the qualification that makes it conditional. Relation endpoints must be identifiable in the capsule or $K_t(a)$; support already present in $K_t(a)$ need not be repeated. A pointer locates evidence but does not by itself establish what that evidence verifies.

Let $I \subseteq E_t$ be the selected records, $L \subseteq R_t$ the conveyed relations, and η their scope, status, and provenance annotations. Admission uses the complete tentative serialization:

$$h = \mathrm{Render}(I, L, \eta), \qquad \mathrm{cost}(h) \leq B. \tag{7}$$

The cost includes field names, relation statements, evidence pointers, and shared annotations as actually rendered. A record that cannot fit with its necessary qualifications is skipped intact, and scanning continues. The renderer groups admitted content by type. For n candidates with precomputed features, ordering costs $O(n \log n)$ and the scan visits each candidate once; extraction, relation handling, and serialization incur additional work.

Dispatch checks. The combined context $(K_t(a), h)$ is checked against configured slot requirements, known constraints, verification records, and open issues, as well as the token budget. A required item omitted despite being available in the graph calls for reselection; missing or insufficient evidence in the graph calls for $\mathcal{R}$. Reselection must restore the affected requirement with its qualifications, rather than repeat an unchanged ranking. A pending check is admissible when performing that check is the assigned step, but cannot serve as a passing result for an action that requires verification. If the checks cannot be satisfied within the budget and revision limits, the handoff is deferred or sent to the workflow's configured fallback. These checks use runtime observations and task templates; the independent reference annotations in Appendix B.1 are reserved for evaluation. Passing the checks means that no violation was found among the represented requirements; it does not establish complete handoff validity.

3.4 FEEDBACK-GUIDED REPAIR

Repair is triggered when a dispatch check or execution feedback identifies an affected slot that needs additional information. An observed failure already containing the relevant condition and evidence updates the graph directly. For example, a failed test on v_2 contradicts a completion claim about v_2 without changing the earlier result for v_1; a request is needed only if the failed condition, log, or action scope is missing.

Request and source selection. A request specifies the slot key, version or condition, and missing field or evidence. Provenance and relation links identify the source: missing constraints are sought from the planner or task specification, outdated results from the verifier or tool log, conflicts from the producers of the incompatible records, and missing justification from the original sender or evidence source. When the record is absent, the configured slot requirement supplies the expected field and responsible role or artifact. The request takes the form of a targeted question or a tool re-check.

Acceptance and stopping. Responses undergo the same schema and reconciliation rules as other observations. A repair closes an information gap only when the requested field or evidence is supplied for the relevant scope and resolves the identified defect. A response that merely repeats a claim leaves its support unresolved. The revised state is checked again before dispatch. Requests stop when the requested information resolves the defect, the configured retry limit is reached, or the remaining execution budget is exhausted. Requirements still preventing dispatch remain explicit and invoke the same deferral or fallback rule. All attempted requests count toward the task-level cost in Equation (2). Appendix A summarizes how these checks, requests, and fallbacks are interleaved in the control flow.

4 EXPERIMENTS

In this section, we evaluate validity-aware state transfer on software repair and constructed multi-role handoff tasks. We first describe the experimental setup, then report end-task results, analyses of state transfer and repair, and component ablations.

4.1 EXPERIMENTAL SETUPS

Datasets and metrics. We evaluate on SWE-bench Lite (Jimenez et al., 2024), which contains 300 software-repair instances, and M2CT-Bench (Multi-Scenario Multi-Agent Context Transfer Benchmark), which we construct for software repair, knowledge analysis, and enterprise workflows. M2CT-Bench comprises 75 base scenarios, 375 context-stress scenarios, and 200 mechanism tasks for budget analysis, state-transfer diagnostics, and ablations, respectively. We report the official resolution rate, token use, estimated cost, and repeated validation attempts on SWE-bench Lite. M2CT-Bench measures state retention, stale-state identification, and recovery from handoff errors. Annotation and metric details are provided in Appendix B.

Models. We use two actor backbones to test whether the benefit of validity-aware state transfer holds across model capability: GPT-5.4 as the strong model and Qwen3.5-4B as the weak model. GPT-5.4 is called through the official API, whereas Qwen3.5-4B is deployed locally on a single NVIDIA RTX 5090 with a maximum context of 32k tokens and a sampling temperature of 0.9. The SWE-bench Lite comparison is run under both backbones, and the ablation and diagnostic experiments use GPT-5.4. Within each controlled comparison, all methods share the actor model, task inputs, role workflow, and tool interfaces. Because the two backbones are served under different configurations, we treat them as two capability regimes rather than as a controlled ablation of backbone choice.

Compared methods. We compare against five baselines: **AutoGen broadcast** (Wu et al., 2023) (full history), **Sliding window** (recent messages), **Summary** (free-form text), **Structured summary** (global JSON state), and **Vector memory** (similarity-based retrieval). We also include the published GPT-5.4 results of Spell and Codex CLI (O'Connor, 2026) as external references, since they use different execution harnesses.

Table 1: SWE-bench Lite results on 300 instances. k denotes 10^3 tokens. Cost per solved instance includes resources spent on unsuccessful runs. Bold marks the best results.

Method	Solved	Rate (%)	Rework ↓	Tokens/solved (k) ↓	USD/solved ↓
GPT-5.4: single-agent results reported by O'Connor (2026)					
Spell (low)	153/300	51.00	-	485.62	-
Spell (medium)	171/300	57.00	-	300.41	≈1.058
Spell (high)	171/300	57.00	-	288.07	-
Codex CLI (low)	170/300	56.70	-	788.76	-
Codex CLI (medium)	172/300	57.30	-	906.74	≈2.376
Codex CLI (high)	185/300	61.67	-	1192.81	-
GPT-5.4: multi-agent results					
Vector memory	122/300	40.67	153	407.04	1.163
Structured summary	82/300	27.33	87	471.68	1.505
AutoGen broadcast	69/300	23.00	79	356.11	1.053
Summary	68/300	22.67	68	418.12	1.305
Sliding window	43/300	14.33	52	345.83	1.056
Validity-aware transfer	**190/300**	**63.33**	**40**	**217.34**	**0.668**
Qwen3.5-4B: multi-agent results					
Vector memory	28/300	9.33	98	262.13	-
Structured summary	18/300	6.00	102	398.15	-
AutoGen broadcast	25/300	8.33	149	601.72	-
Summary	22/300	7.33	89	314.62	-
Sliding window	12/300	4.00	73	550.02	-
Validity-aware transfer	**97/300**	**32.33**	**66**	**109.12**	-

4.2 Main Results

Table 1 shows that validity-aware transfer resolves 190 of 300 SWE-bench Lite instances (63.33%) with GPT-5.4, exceeding vector memory (40.67%), the strongest evaluated multi-agent baseline by resolution rate, by 22.67 percentage points. Compared with this baseline, total token use falls from 49.659M to 41.295M (a 16.84% reduction), tokens per resolved instance from 407.04k to 217.34k (46.60%), and repeated validation attempts from 153 to 40; estimated cost per resolved instance decreases from \$1.163 to \$0.668. Sliding window uses fewer total tokens but resolves only 43 instances, illustrating the need to consider cost alongside success. Our resolution rate also exceeds the reported high-effort results of Codex CLI (61.67%) and Spell (57.00%).

The state-level diagnostics below suggest why. Baselines retain a substantial share of task content but lose validity structure, so a receiver cannot tell which facts are still current and ends up revalidating work that is already settled; this is consistent with the 153 repeated validation attempts recorded for vector memory against 40 for our method. Typed slots carry that structure inside the same budget, so the receiver can act on state it is able to check and request only the fields that are genuinely missing. The advantage also grows as the backbone weakens: under Qwen3.5-4B the same protocol resolves 97 of 300 instances (32.33%) against 28 (9.33%) for the strongest baseline, a 3.5-fold gap compared with 1.6-fold for GPT-5.4, since a smaller model has less capacity to reconstruct missing or superseded state from its own context.

4.3 Diagnostic Analyses

State retention and budget sensitivity. On the 375 M2CT-Bench stress scenarios, validity-aware transfer improves state recall to 0.4396, stale-state identification to 0.9350, and relation coverage to 0.7501, compared with the strongest baseline values of 0.2007, 0.6328, and 0.0533, respectively (Figure 3(a)). Evidence coverage improves only slightly, from 0.1251 to 0.1279, so the largest gains concern temporal status and relations, with substantial state and evidence still missing. Under a shared 512-token budget on the 75 base scenarios (Figure 3(b)), critical-state recall reaches 0.4794 versus 0.4027 for summary, while stale inclusion falls to 0.0269 versus 0.0286 for struc-

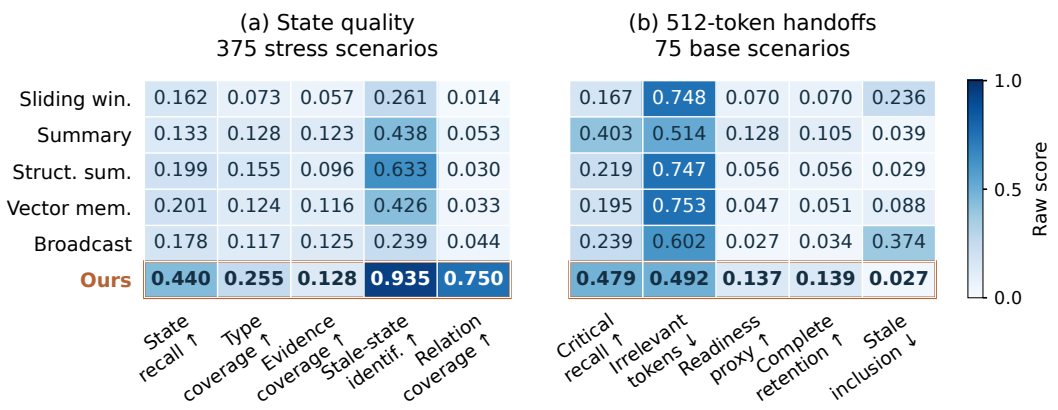

Figure 3: State-transfer diagnostics with GPT-5.4 on M2CT-Bench. (a) State quality on 375 stress scenarios. (b) Receiver-conditioned transfer at 512 tokens on 75 base scenarios. Cell labels and colors show raw scores; arrows indicate the preferred direction. Values are rounded to three decimals. Orange outlines mark validity-aware transfer (Ours).

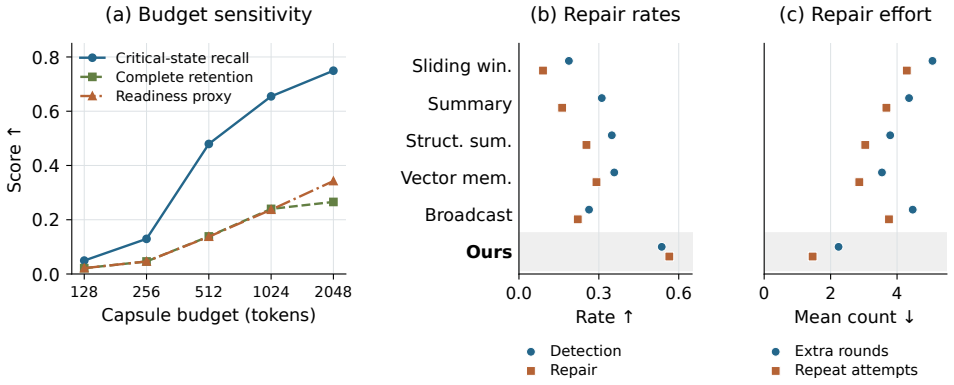

Figure 4: Budget sensitivity and feedback-guided repair with GPT-5.4. (a) Capsule-budget sweep on 75 base scenarios. (b) Explicit detection and repair success on 375 stress scenarios, scored separately. (c) Mean extra rounds and repeat repair attempts on the same stress scenarios. Shaded rows mark our method. Points show reported aggregates.

tured summary. The action-readiness proxy improves only from 0.1282 to 0.1374; this measure checks retained state against receiver requirements rather than executing the next action. Increasing our capsule budget from 128 to 2,048 tokens raises critical-state recall from 0.0495 to 0.7494, but complete retention reaches only 0.2656 and the readiness proxy 0.3426 (Figure 4(a)). Thus, larger budgets recover more individual items while many handoffs still lack the combination required by the receiver. Appendix B.2 defines the metrics.

Error recovery and transfer across domains. On the 375 stress scenarios, validity-aware transfer achieves 0.5361 explicit error detection and 0.5647 repair success, compared with 0.3579 and 0.2911 for vector memory; mean additional rounds fall from 3.5455 to 2.2393, and repeat repair attempts from 2.8644 to 1.4578 (Figure 4(b,c)). Detection and repair are scored separately because a correction need not include an explicit error declaration. These results support the use of slots and provenance to guide recovery, although many errors remain unrepaired. Figure 5 examines evidence and approval state beyond software repair: on 225 knowledge-analysis scenarios, fact retention and citation precision reach 0.3978 and 0.4637, compared with the best baseline values of 0.1897 and 0.3148, while unsupported conclusions decrease to 0.2387.

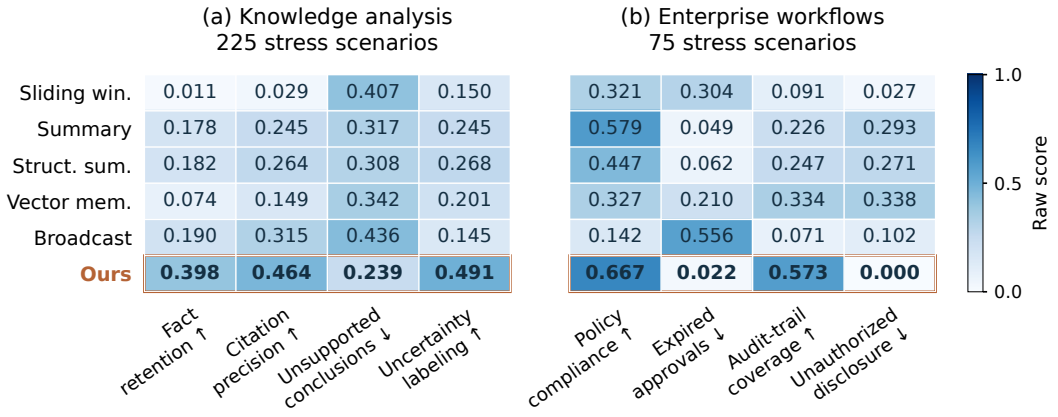

Figure 5: Cross-domain results with GPT-5.4 on the M2CT-Bench stress scenarios: (a) knowledge analysis and (b) enterprise workflows. Labels and colors show raw scores rounded to three decimals; arrows specify whether higher or lower values are preferred. Orange outlines mark our method.

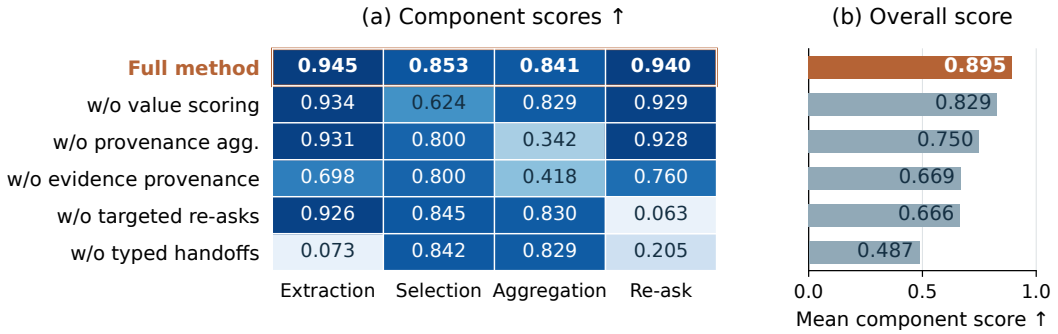

Figure 6: GPT-5.4 ablations on 200 M2CT-Bench mechanism tasks, with 50 tasks per category. (a) Component scores on a fixed 0–1 color scale. (b) Their arithmetic mean. Higher is better. Orange marks the full method. Scores measure component behavior, not SWE-bench resolution; scoring definitions are in Appendix B.3.

4.4 Ablation Study

Figure 6 compares five ablations on the same 200 mechanism tasks. Removing value scoring reduces selection from 0.853 to 0.624 while leaving the other component scores close to the full method, indicating its role in prioritizing records under budget. Removing provenance-aware aggregation lowers aggregation from 0.841 to 0.342; removing evidence provenance lowers it to 0.418 and extraction from 0.945 to 0.698, indicating that provenance supports both state identification and reconciliation. Without targeted re-asking, the re-ask score drops from 0.940 to 0.063 while other scores remain at least 0.830, separating state maintenance from recovery of missing information. Removing typed handoffs produces the largest overall decrease, from 0.895 to 0.487, with extraction and re-asking falling to 0.073 and 0.205.

5 Conclusion

In this paper, we study how handoffs in multi-agent LLM systems can preserve a task's facts while losing the constraints, verification status, and dependencies that make them safe to act on. We formulate budgeted handoff as receiver-conditioned state projection and propose validity-aware state transfer, a training-free method that maintains a typed state graph with provenance, evidence, and lifecycle status, charges each item's qualifications to the same budget as the item, and repairs affected slots from execution feedback. It resolves 190 of 300 SWE-bench Lite instances (63.33% versus

40.67% for the strongest baseline) at 217.34k rather than 407.04k tokens per resolved instance, and identifies stale state at 0.9350 on controlled handoffs. Retention nonetheless remains partial, error repair succeeds on 56.47% of failures, and selecting the combination a receiver jointly needs rather than the items that rank highest remains the central open problem.

6 AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on the methodology and experiments, to assist with implementing the method and its evaluation, to help interpret results, and to assist with translation. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Anthropic. Effective harnesses for long-running agents. Anthropic Engineering, 2025. URL https://www.anthropic.com/engineering/effective-harnesses-for-long-running-agents.

Pratyay Banerjee and Ankit Chadha. Routed graph handoff: Adaptive format selection for multi-agent LLM delegation. *arXiv preprint arXiv:2608.25277*, 2026. URL https://arxiv.org/abs/2608.25277.

Amine Barrak. Traceability and accountability in role-specialized multi-agent LLM pipelines. *arXiv preprint arXiv:2510.07614*, 2025. URL https://arxiv.org/abs/2510.07614.

Shiyang Chen. Governance decay: How context compaction silently erases safety constraints in long-horizon LLM agents. *arXiv preprint arXiv:2606.22528*, 2026. URL https://arxiv.org/abs/2606.22528.

Runzhi Deng, Yiming Zhong, Fang Zhao, and Pan Zhou. Do not restart: Residual completion for stateful agent handoffs. *arXiv preprint arXiv:2609.13800*, 2026. URL https://arxiv.org/abs/2609.13800.

Roy Ganz, Mor Shpigel Nacson, Adi Kalyanpur, and Ron Litman. The handoff tax: Continuing non-native trajectories in LLM agents. *arXiv preprint arXiv:2608.24358*, 2026. URL https://arxiv.org/abs/2608.24358.

Yibo Hu. Silence is endorsement: Verification-status laundering in LLM agent pipelines. *arXiv preprint arXiv:2609.20211*, 2026. URL https://arxiv.org/abs/2609.20211.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://aclanthology.org/2023.emnlp-main.825/.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.06770.

Taejong Joo, Shu Ishida, Ivan Sosnovik, Bryan Lim, Sahand Rezaei-Shoshtari, Adam Gaier, and Robert Giaquinto. Graph of agents: Principled long context modeling by emergent multi-agent collaboration. *arXiv preprint arXiv:2509.21848*, 2025. URL https://arxiv.org/abs/2509.21848.

Dipesh KC and Anjila Budathoki. Handoff debt: The rediscovery cost when coding agents take over interrupted tasks. *arXiv preprint arXiv:2606.02875*, 2026. URL https://arxiv.org/abs/2606.02875.

Bowen Li and Guojun Wang. Governance at the boundary: How agent decomposition degrades policy compliance. *arXiv preprint arXiv:2608.16055*, 2026. URL https://arxiv.org/abs/2608.16055.

Jiayu Li, Enpei Zhang, Dawei Zhou, Elynn Chen, and Yujun Yan. Learning to hand off: Provably convergent workflow learning under interface constraints. *arXiv preprint arXiv:2605.19140*, 2026a. URL https://arxiv.org/abs/2605.19140.

Xiaoyang Li, Yiqi Wang, Haohui Lu, Zhi Chen, Mo Li, Pingan Song, Mingkai Zheng, and Taotao Cai. Memtx: Transactional belief commit for stateful agent memory. *arXiv preprint arXiv:2607.23929*, 2026b. URL https://arxiv.org/abs/2607.23929.

Tobias Lindenbauer, Igor Slinko, Ludwig Felder, Egor Bogomolov, and Yaroslav Zharov. The complexity trap: Simple observation masking is as efficient as LLM summarization for agent context management. *arXiv preprint arXiv:2508.21433*, 2025. URL https://arxiv.org/abs/2508.21433.

Jiawei Liu, Peiyi Han, Yuntao Lu, Su Zheng, Fengyu Yan, and Bei Yu. Agentic electronic design automation: A handoff perspective. *arXiv preprint arXiv:2606.19795*, 2026. URL https://arxiv.org/abs/2606.19795.

Xuan Liu and Jingbin Qian. Reach or solve? attributing agentic RL gains with checkpoint handoffs. *arXiv preprint arXiv:2609.19636*, 2026. URL https://arxiv.org/abs/2609.19636.

Yanki Margalit, Nurit Cohen-Inger, Erni Avram, Ran Taig, and Oded Margalit. Governed shared memory for multi-agent llm systems. *arXiv preprint arXiv:2606.24535*, 2026. URL https://arxiv.org/abs/2606.24535.

Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, Chenlin Zhou, Jiayi Mao, Tianze Xia, Jiafeng Guo, and Shenghua Liu. A survey of context engineering for large language models. *arXiv preprint arXiv:2507.13334*, 2025. URL https://arxiv.org/abs/2507.13334.

Luke J. O'Connor. Self-programmed execution for language-model agents. *arXiv preprint arXiv:2605.06898*, 2026. URL https://arxiv.org/abs/2605.06898.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 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, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.810. URL https://aclanthology.org/2024.acl-long.810/.

Ke Rui, Yushen Zuo, Jiawei Wang, Haoran Jia, Jinming Ma, Weitao Zhou, and Minglei Li. Diagnosing semantic handoff failures in agent-orchestrated vision-language-action skill composition. *arXiv preprint arXiv:2607.06256*, 2026. URL https://arxiv.org/abs/2607.06256.

Kaiwen Shi, Zheyuan Zhang, Han Bao, Colby Nelson, and Yanfang Ye. Confidence laundering in agent systems: Why uncertainty needs a latent carrier. *arXiv preprint arXiv:2606.20662*, 2026. URL https://arxiv.org/abs/2606.20662.

Prabhjot Singh and Bhushan Pawar. The hallucination snowball: Modeling error propagation as state transitions in multi-agent LLM pipelines. *arXiv preprint arXiv:2608.14588*, 2026. URL https://arxiv.org/abs/2608.14588.

Divyansh Srivastava, Shreya Ghosh, Anshul Verma, and Rajkumar Buyya. Agentic AI-based framework for mitigating premature diagnostic handoff and silent hallucination in healthcare applications. *arXiv preprint arXiv:2606.18068*, 2026. URL https://arxiv.org/abs/2606.18068.

Yiheng Sun, Huifei Wang, Yancheng Zhu, Zhenyu Li, Zebin Zhao, and Yifan Yuan. When "must" becomes "maybe": Constraint weakening in LLM agent workflows. *arXiv preprint arXiv:2608.24569*, 2026. URL https://arxiv.org/abs/2608.24569.

Rui Tang, Qiangqiang Liu, Yichi Zhang, Youwei Yang, Xi Chen, and Chen Dong. Context is not authority: Structured runtime governance for financial market agents. *arXiv preprint arXiv:2608.09025*, 2026. URL https://arxiv.org/abs/2608.09025.

Yichen Tang, Weihang Su, Yujia Zhou, Yiqun Liu, Min Zhang, Shaoping Ma, and Qingyao Ai. Augmenting multi-agent communication with state delta trajectory. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10219–10240, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.518. URL https://aclanthology.org/2025.emnlp-main.518/.

Hayder Tirmazi, Sam Markelon, Allison Bishop, and Michael Mitzenmacher. Context compaction theory. *arXiv preprint arXiv:2608.01326*, 2026. URL https://arxiv.org/abs/2608.01326.

Natalia Trukhina and Vadim Vashkelis. Compress the context, keep the commitments: A formal framework for verifiable LLM context compression. *arXiv preprint arXiv:2605.17304*, 2026. URL https://arxiv.org/abs/2605.17304.

Sergey Volkov, Yang Li, and Ye Luo. Statefuse: Deterministic conflict-preserving memory for multi-agent systems. *arXiv preprint arXiv:2607.05844*, 2026. URL https://arxiv.org/abs/2607.05844.

Song Wang, Zhen Tan, Zihan Chen, Shuang Zhou, Tianlong Chen, and Jundong Li. Anymac: Cascading flexible multi-agent collaboration via next-agent prediction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 11555–11567, Suzhou, China, 2025a. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.584. URL https://aclanthology.org/2025.emnlp-main.584/.

Yian Wang, Agam Goyal, Eshwar Chandrasekharan, and Hari Sundaram. Facts without rules: Boundary metadata collapse in multi-agent LLM handoffs. *arXiv preprint arXiv:2608.29028*, 2026. URL https://arxiv.org/abs/2608.29028.

Zhexuan Wang, Yutong Wang, Xuebo Liu, Liang Ding, Miao Zhang, Jie Liu, and Min Zhang. Agentdropout: Dynamic agent elimination for token-efficient and high-performance llm-based multi-agent collaboration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24013–24035, Vienna, Austria, 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.1170. URL https://aclanthology.org/2025.acl-long.1170/.

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. *arXiv preprint arXiv:2308.08155*, 2023. URL https://arxiv.org/abs/2308.08155.

Hancheng Ye, Zhengqi Gao, Mingyuan Ma, Qinsi Wang, Yuzhe Fu, Ming-Yu Chung, Yueqian Lin, Zhijian Liu, Jianyi Zhang, Danyang Zhuo, and Yiran Chen. Kvcomm: Online cross-context kv-cache communication for efficient llm-based multi-agent systems. *arXiv preprint arXiv:2510.12872*, 2025. URL https://arxiv.org/abs/2510.12872.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. Cut the crap: An economical communication pipeline for llm-based multi-agent systems. In *International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=LkzuPorQ5L.

Ruijia Zhang, Xinyan Zhao, Ruixiang Wang, Sigen Chen, Guibin Zhang, An Zhang, Kun Wang, and Qingsong Wen. Safesieve: From heuristics to experience in progressive pruning for llm-based multi-agent communication. *arXiv preprint arXiv:2508.11733*, 2025b. URL https://arxiv.org/abs/2508.11733.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan O. Arik. Chain of agents: Large language models collaborating on long-context tasks. In *Advances in Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=LuCLf4BJsr.

Heng Zhou, Lian Zhang, Yutao Fan, Tiancheng He, Siki Chen, Hejia Geng, Philip Torr, and Zhenfei Yin. Latticemind: A conflict-aware memory primitive for multi-agent systems. *arXiv preprint arXiv:2608.08236*, 2026. URL https://arxiv.org/abs/2608.08236.

A STATE-TRANSFER PROCEDURE

A.1 COMPLETE PROCEDURE

Algorithm 1 specifies the control flow for capsule reselection, information requests, dispatch checks, and fallback across successive handoffs.

Algorithm 1: Validity-aware state transfer at successive handoffs

Input: Initial observations; workflow-provided receivers, budgets, and revision limits
Initialize $\mathcal{S}$ by applying U to the initial observations;
while *the task continues within its execution budget* **do**
  Obtain receiver a, next-step context x, existing context $K(a)$, and budget B;
  Construct $h \leftarrow \Gamma(\mathcal{S}, x, a, K(a), B)$ and check dispatch requirements;
  while *a check fails and revision limits and remaining budget permit* **do**
    if *the graph has sufficient information to correct the capsule* **then**
      Reselect to restore missing requirements or remove invalid premises; re-render;
    else
      Request the affected field or evidence using $\mathcal{R}$; update $\mathcal{S}$ through U;
      Reconstruct h from the updated graph;
    Recheck complete token cost, qualifications, and required slots in $(K(a), h)$;
  if *all dispatch checks pass* **then**
    Execute the assigned step with $(K(a), h)$; collect observations and feedback;
    Update $\mathcal{S}$ through U;
    if *a localized defect still needs information and repair limits and budget permit* **then**
      Request the missing information using $\mathcal{R}$; update $\mathcal{S}$ through U;
  else
    Invoke the configured fallback or defer the handoff; retain unresolved requirements;

B M2CT-BENCH ANNOTATIONS AND METRIC DEFINITIONS

B.1 REFERENCE ANNOTATIONS

Twenty annotators with software, knowledge-analysis, or multi-agent experience labeled critical state items, types, evidence sources, receiver requirements, temporal validity, and conflicts. They first aligned the rubric on 20 examples excluded from the benchmark. Three annotators independently labeled each example; a fourth adjudicated disagreements, and 30% of annotations were rechecked.

B.2 DIAGNOSTIC METRIC DEFINITIONS

State retention, lifecycle, and relations. On the 375 stress scenarios, state recall is the fraction of reference critical items retained. Type coverage counts reference items assigned to the correct type; evidence coverage counts correctly preserved reference evidence associations. Stale-state identification counts reference stale items correctly filtered or marked, and relation coverage counts reference support, conflict, and replacement relations correctly expressed. All five are calculated per scenario and averaged. Evidence coverage uses reference associations as its denominator, rather than emitted pointers.

Let $\mathcal{D}$ be the 375 context-stress scenarios and, for a scenario d, let E_d^* and E_d be the reference critical items and those retained by the evaluated state, $\tau_d^*(e)$ and $\tau_d(e)$ the reference and assigned type of item e, A_d^* and A_d the reference and correctly preserved evidence associations, R_d^* and R_d the reference and correctly expressed relations, and $E_d^{*,\mathrm{st}}$ and E_d^{st} the reference stale items and those correctly filtered or marked. Each quantity is computed per scenario and averaged over $\mathcal{D}$:

$$\begin{aligned}
\mathrm{Recall} &= \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}}\frac{|E_d\cap E_d^*|}{|E_d^*|}, \quad \mathrm{Type} = \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}}\frac{|\{e\in E_d\cap E_d^* : \tau_d(e)=\tau_d^*(e)\}|}{|E_d^*|},\\
\mathrm{Evid} &= \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}}\frac{|A_d\cap A_d^*|}{|A_d^*|}, \quad \mathrm{Stale} = \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}}\frac{|E_d^{\mathrm{st}}|}{|E_d^{*,\mathrm{st}}|},\\
\mathrm{Rel} &= \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}}\frac{|R_d\cap R_d^*|}{|R_d^*|}.
\end{aligned} \tag{8}$$

Receiver-conditioned transfer and budget. On the 75 base scenarios, critical-state recall uses the receiver's required items as its denominator. Irrelevant-token fraction counts transmitted tokens unrelated to those requirements; complete retention is the fraction of handoffs retaining all critical items; stale inclusion counts selected stale items relative to the scenario's stale items. A handoff passes the action-readiness proxy when it retains all high-priority items, covers at least 75% of the receiver's required state, and includes no stale item relevant to that role. This rule evaluates supplied context, not an independently executed action. Scores are averaged across scenarios and receivers. The fixed-budget comparison uses 512 tokens; the sensitivity analysis varies our budget from 128 to 2,048 tokens.

Let $\mathcal{U}$ collect the handoff units of the 75 base scenarios, each pairing a scenario d with a receiving role a and its reference required items $Q_d(a)$, and let $\hat{E}_{d,a}$ be the state presented to a by the capsule together with its existing context. With $c_{d,a}$ the transmitted tokens and $c_{d,a}^{\mathrm{irr}}$ those unrelated to $Q_d(a)$, $Q_d^{\mathrm{hi}}(a)$ the high-priority requirements, E_d^{st} the scenario's stale items, $E_d^{\mathrm{st,rel}}(a)\subseteq E_d^{\mathrm{st}}$ those relevant to the receiving role, and $\hat{E}_{d,a}^{\mathrm{st}}\subseteq E_d^{\mathrm{st}}$ the stale items still transmitted:

$$\begin{aligned}
\mathrm{CritRecall} &= \frac{1}{|\mathcal{U}|}\sum_{(d,a)\in\mathcal{U}}\frac{|\hat{E}_{d,a}\cap Q_d(a)|}{|Q_d(a)|}, \qquad \mathrm{IrrTok} = \frac{1}{|\mathcal{U}|}\sum_{(d,a)\in\mathcal{U}}\frac{c_{d,a}^{\mathrm{irr}}}{c_{d,a}},\\
\mathrm{Complete} &= \frac{1}{|\mathcal{U}|}\sum_{(d,a)\in\mathcal{U}}\mathbf{1}\left[Q_d(a)\subseteq\hat{E}_{d,a}\right], \quad \mathrm{StaleInc} = \frac{1}{|\mathcal{U}|}\sum_{(d,a)\in\mathcal{U}}\frac{|\hat{E}_{d,a}^{\mathrm{st}}|}{|E_d^{\mathrm{st}}|},
\end{aligned} \tag{9}$$

The action-readiness proxy is the indicator

$$\mathrm{Ready}_{d,a} = \mathbf{1}\left[Q_d^{\mathrm{hi}}(a) \subseteq \hat{E}_{d,a} \ \wedge \ \frac{|\hat{E}_{d,a} \cap Q_d(a)|}{|Q_d(a)|} \geq 0.75 \ \wedge \ \hat{E}_{d,a}^{\mathrm{st}} \cap E_d^{\mathrm{st,rel}}(a) = \emptyset\right], \tag{10}$$

whose reported value averages it over $\mathcal{U}$.

Error detection and repair. The 375 stress scenarios include omitted constraints, reuse of stale evidence, unverified hypotheses treated as facts, decisions detached from their support, and missing current verification results. Explicit detection requires a role to identify the erroneous state or missing slot and associate it with a source or evidence. Repair succeeds when the state is corrected, replaced, invalidated, or supplemented as appropriate and the subsequent action satisfies the reference answer. Detection and repair are scored separately over error scenarios: an agent can correct a problem without explicitly declaring it, so repair success need not be bounded by explicit detection. Additional rounds and repeat repair attempts are averaged over scenarios.

Let $\mathcal{D}_{\mathrm{err}}$ be the error scenarios, and let D_d and P_d be the indicators that the erroneous or missing state is explicitly identified together with its source or evidence and that the state is corrected, replaced, invalidated, or supplemented so that the subsequent action matches the reference answer. With r_d the additional rounds beyond the reference count and, over the scenarios $\mathcal{D}_{\mathrm{err}}^{\mathrm{fail}}$ whose initial repair failed, ρ_d the repair attempts that followed:

$$\begin{aligned} \mathrm{Detect} &= \frac{1}{|\mathcal{D}_{\mathrm{err}}|}\sum_{d\in\mathcal{D}_{\mathrm{err}}} D_d, \quad \mathrm{Repair} = \frac{1}{|\mathcal{D}_{\mathrm{err}}|}\sum_{d\in\mathcal{D}_{\mathrm{err}}} P_d, \\ \mathrm{Rounds} &= \frac{1}{|\mathcal{D}_{\mathrm{err}}|}\sum_{d\in\mathcal{D}_{\mathrm{err}}} r_d, \quad \mathrm{Repeat} = \frac{1}{|\mathcal{D}_{\mathrm{err}}^{\mathrm{fail}}|}\sum_{d\in\mathcal{D}_{\mathrm{err}}^{\mathrm{fail}}} \rho_d. \end{aligned} \tag{11}$$

Cross-domain metrics. The 225 knowledge-analysis and 75 enterprise-workflow stress scenarios use different metrics. In knowledge analysis, fact retention covers reference facts and constraints, citation precision measures correct source references among emitted citations, unsupported-conclusion rate measures claims lacking factual or citation support, and uncertainty labeling checks whether conclusions are marked verified, pending, or conflicting. In enterprise workflows, policy compliance evaluates actions subject to permissions, approval, or business rules; expired-approval reuse counts stale approvals still adopted; audit-trail coverage measures retained reference audit elements; and unauthorized disclosure measures sensitive state sent to a role without permission. Knowledge scores are averaged over scenarios, and enterprise scores over the corresponding handoffs and scenarios.

For knowledge analysis, let $\mathcal{D}_{\mathrm{K}}$ be the 225 stress scenarios in that domain, F_d^* and F_d the reference and retained facts or constraints, $\hat{C}_d$ the emitted citations and $\hat{C}_d^{\mathrm{ok}} \subseteq \hat{C}_d$ those that are correct, L_d the emitted conclusions, $U_d \subseteq L_d$ those lacking factual or citation support, and $Z_d \subseteq L_d$ those correctly labeled verified, pending, or conflicting:

$$\begin{aligned} \mathrm{Fact} &= \frac{1}{|\mathcal{D}_{\mathrm{K}}|}\sum_{d\in\mathcal{D}_{\mathrm{K}}} \frac{|F_d \cap F_d^*|}{|F_d^*|}, \qquad \mathrm{Cite} = \frac{1}{|\mathcal{D}_{\mathrm{K}}|}\sum_{d\in\mathcal{D}_{\mathrm{K}}} \frac{|\hat{C}_d^{\mathrm{ok}}|}{|\hat{C}_d|}, \\ \mathrm{Unsup} &= \frac{1}{|\mathcal{D}_{\mathrm{K}}|}\sum_{d\in\mathcal{D}_{\mathrm{K}}} \frac{|U_d|}{|L_d|}, \qquad \mathrm{Uncert} = \frac{1}{|\mathcal{D}_{\mathrm{K}}|}\sum_{d\in\mathcal{D}_{\mathrm{K}}} \frac{|Z_d|}{|L_d|}. \end{aligned} \tag{12}$$

For enterprise workflows, let $\mathcal{U}_{\mathrm{E}}$ be the handoff units of the 75 stress scenarios in that domain, $\Pi_{d,a}$ the actions subject to permissions, approval, or business rules and $\Pi_{d,a}^{\mathrm{ok}}$ those that comply, S_d^* the reference approvals and $S_{d,a} \subseteq S_d^*$ the stale approvals still adopted, T_d^* and $T_{d,a}$ the reference and retained audit elements, and $N_{d,a}^*$ the sensitive reference items that role a is not permitted to receive, with $N_{d,a} \subseteq N_{d,a}^*$ those actually transmitted:

$$\begin{aligned} \mathrm{Policy} &= \frac{1}{|\mathcal{U}_{\mathrm{E}}|}\sum_{(d,a)\in\mathcal{U}_{\mathrm{E}}} \frac{|\Pi_{d,a}^{\mathrm{ok}}|}{|\Pi_{d,a}|}, \qquad \mathrm{Expired} = \frac{1}{|\mathcal{U}_{\mathrm{E}}|}\sum_{(d,a)\in\mathcal{U}_{\mathrm{E}}} \frac{|S_{d,a}|}{|S_d^*|}, \\ \mathrm{Audit} &= \frac{1}{|\mathcal{U}_{\mathrm{E}}|}\sum_{(d,a)\in\mathcal{U}_{\mathrm{E}}} \frac{|T_{d,a} \cap T_d^*|}{|T_d^*|}, \quad \mathrm{Disclose} = \frac{1}{|\mathcal{U}_{\mathrm{E}}|}\sum_{(d,a)\in\mathcal{U}_{\mathrm{E}}} \frac{|N_{d,a}|}{|N_{d,a}^*|}. \end{aligned} \tag{13}$$

B.3 COMPONENT SCORING

The 200 ablation tasks are divided equally among four mechanism categories. Each task is scored against its reference annotations, then scores are averaged within the corresponding category. Set matching for state items, evidence, and relations uses the harmonic mean of precision and recall. The extraction score averages item identification, type-slot assignment, status assignment, and evidence-pointer matching. The selection score combines critical-state recall under budget and ranking quality. The aggregation score combines active-state identification, relation matching, and stale-state replacement. The re-ask score combines response parsing, target identification, and the match of supplemented items, types, and evidence.

With P and R the precision and recall of an emitted set $\hat{X}$ against its reference set X^*, set matching uses

$$F_1(\hat{X}, X^*) = \frac{2PR}{P+R} = \frac{2\,|\hat{X} \cap X^*|}{|\hat{X}| + |X^*|}, \qquad \text{Overall} = \frac{1}{4}\left(\text{Ext} + \text{Sel} + \text{Agg} + \text{ReAsk}\right), \tag{14}$$

where each of Ext, Sel, Agg, and ReAsk averages its named subscores, and $\hat{X}$ and X^* stand for a set of state items, evidence links, or relations. The stress-scenario state metrics use the reference as the denominator. In particular, the source results' "type accuracy," "evidence-pointer accuracy," and "relation accuracy" count correctly preserved reference items or associations.

C Detailed Related Work

This appendix provides the complete analysis of the reference literature, organized by how each group relates to the problem studied here.

Efficient communication, long-context decomposition, and compression. Multi-agent frameworks exchange messages between role-specialized agents (Wu et al., 2023; Qian et al., 2024), and much subsequent work optimizes the communication graph rather than the content of a transfer, by pruning agents, edges, or messages (Zhang et al., 2025a; Wang et al., 2025b; Zhang et al., 2025b) or by predicting the next agent (Wang et al., 2025a). A related line decomposes long inputs across chained or graph-structured agents (Zhang et al., 2024; Joo et al., 2025). On the compression side, prompt compression and context-engineering surveys describe the design space (Jiang et al., 2023; Mei et al., 2025), a controlled comparison finds simple observation masking competitive with LLM summarization (Lindenbauer et al., 2025), and compaction theory relates bounded-length generation to one-way communication complexity and shows that selection and generation can differ (Tirmazi et al., 2026). These methods decide who communicates, how much is retained, and when to compress; they do not represent whether a retained item is still applicable to the receiver's next action.

Handoff representation, state transfer, and memory. Structured interfaces improve cross-role continuation: traceable and auditable handoffs (Barrak, 2025), per-delegation choice between a typed dependency graph and natural language (Banerjee & Chadha, 2026), typed and source-grounded semantic atoms for verifiable compression (Trukhina & Vashkelis, 2026), commitment-constrained residual completion with partial-correctness guarantees (Deng et al., 2026), and workflow learning that bounds the loss induced by the interface representation gap (Li et al., 2026a). Handoff objects have also been catalogued by their producers and consumers in agentic electronic design automation (Liu et al., 2026). Transfer cost and quality have been measured by handing a frozen interrupted task to a successor with different views (KC & Budathoki, 2026), by comparing upgrade and downgrade directions (Ganz et al., 2026), and by separating reaching a state from solving from it (Liu & Qian, 2026). Alternatives to text include communicating state deltas alongside messages (Tang et al., 2025) and reusing KV caches across agents (Ye et al., 2025), and gating on completeness has been used to block premature transfer in clinical workflows (Srivastava et al., 2026). On the store side, LatticeMind records which claim currently wins and why a later update supersedes it (Zhou et al., 2026); StateFuse preserves disagreement under a replicated merge contract with immutable history (Volkov et al., 2026); MemTX treats accepted writes as beliefs that require commitment before they can drive side effects (Li et al., 2026b); and governed shared memory formalizes scoped access, temporal correctness, and policy-controlled propagation for fleets of agents (Margalit et al., 2026). We reuse these ideas about provenance and conflict, but our unit of analysis is not the store: it is the bounded projection of the store that one receiver sees at one boundary, where a correct store can still yield an invalid handoff.

Diagnostics of handoff failure. Several recent studies independently establish that handoffs lose more than content. Boundary-metadata collapse shows that facts survive while the boundary markers governing their use do not (Wang et al., 2026); constraint weakening measures the degradation of blocking conditions into caveats after compression (Sun et al., 2026); governance decay shows that compaction silently erases safety constraints and raises violation rates (Chen, 2026); verification-status laundering shows that removing an "unverified" marker raises risky-action approval (Hu, 2026); confidence laundering argues that uncertainty needs a carrier that survives component boundaries (Shi et al., 2026); and the hallucination snowball models how errors escape successive handoff boundaries (Singh & Pawar, 2026). Deeper decomposition is associated with greater policy-fact decay (Li & Wang, 2026), contextual presence is distinguished from authority to act (Tang et al., 2026), and semantic handoff failures in skill composition show that satisfying a local postcondition can still leave the successor unable to start (Rui et al., 2026). These works diagnose failure surfaces separately; we treat them as independent evidence for the problem and provide a single runtime intervention that maintains lifecycle, scope, and relation state, then measures which failures it closes.