AFTER THE FIX: TRANSFER OF CORRECTED AGENT EXPERIENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Does repairing an episode make its experience a better memory for the next task? We transfer the same failed source before and after accepted repair to a fixed target, alongside independent execution. Our 3,300 runs cover 100 ThinkingBox pairs and the same 100 APEX pairs with and without source-state inheritance, under eleven conditions. ThinkingBox's Full/Skill/Hybrid correction gains are 44/29/32 percentage points, with corrected performance 25/22/18 points above independence; inference weakens at the task-family level. Yet 12 of Full's 15-point larger correction gap over Skill come from worse uncorrected performance, not better corrected memory. Moreover, 22 of Full's 46 upward transitions restore observed baseline success. Neither APEX regime establishes comparable aggregate correction benefits. Action evidence connects workflow gains with reusable obligations and convention conflicts with source-local choices. Text APEX's accepted execution reaches 52% versus its summary's 40%, without robust global/group-level superiority or an established advantage over independence. Smaller handoffs reduce input but increase calls. The value of repairing experience is therefore distinct from the value of reusing it: memory updates require both a previous-version reference and a fresh-start reference.

1 INTRODUCTION

An agent repairs a failed task, receives approval, and handles a related request. Its memory now has two versions: the initial failure and an accepted repair. Should the latter replace the former? A system that fixes yesterday's mistake while applying yesterday's assumptions to today's request has improved its source solution without necessarily improving its memory for the next task.

The value of repairing experience is not the same as the value of reusing it. A correction can expose an obligation, such as opening a ticket before acting and closing it afterward, or establish a local workbook version, valuation convention, or input range. The obligation may survive a changed customer; the choice requires renewed grounding. Acceptance validates a source outcome, not the scope of its reuse.

Iterative refinement improves attempts through feedback (Madaan et al., 2023; Shinn et al., 2023); memories and workflows extend reuse across tasks (Zhao et al., 2024; Wang et al., 2025; Ouyang et al., 2026). We study **the incremental transfer value of updating the same failed experience through repair**. This asks whether correction improves a particular memory, beyond whether successful or failed experience can be useful in general.

We study source A at two points: **U** stops after initial failure; **C** continues through feedback and repair to acceptance. Both supply the same fixed target B through independent executions. C–U asks whether repair improves the memory; C–B-only asks whether using it beats a fresh start. Pairing the same source holds provenance fixed rather than comparing unrelated successful and failed episodes. A large correction gap can reflect weak U, strong C, or both.

We cross this comparison with how experience is retained. Full history, procedural skill, and hybrid representations select different information; a minimal None condition isolates their shared channel. Summary and Execution ask whether the accepted action segment adds value beyond its outcome summary. Across 3,300 runs, ThinkingBox tests workflow reuse, while matched APEX-state and APEX-text cohorts distinguish implemented handoffs with and without source-artifact inheritance.

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.

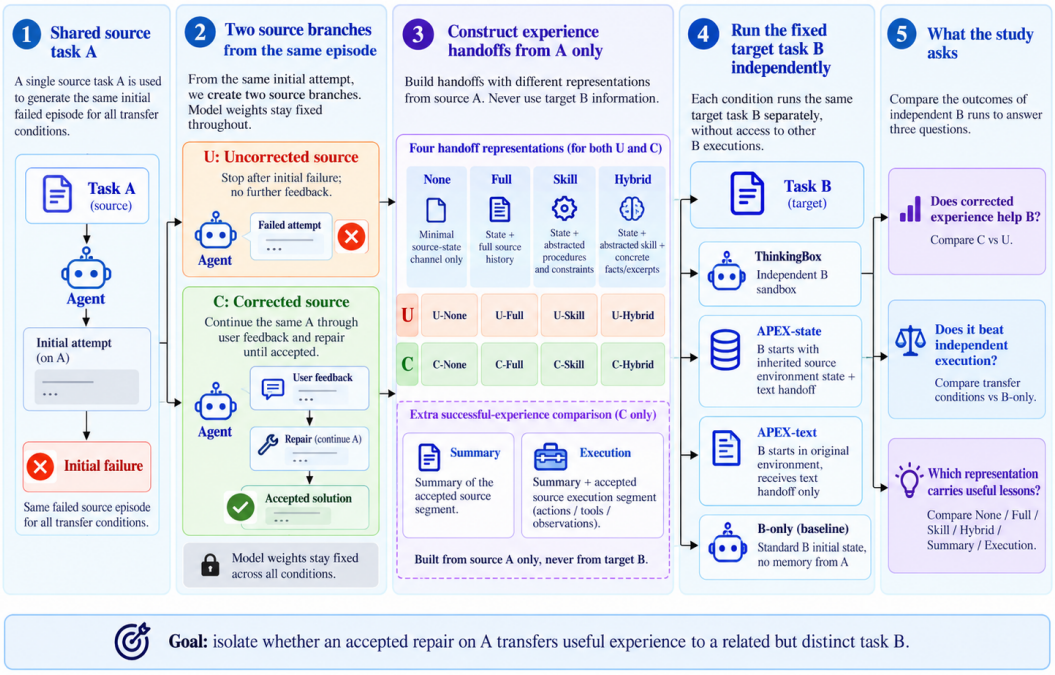


Figure 1: The same failed A branches into U/C; each handoff runs B independently. State and text are separate channels. Table 1 defines all six handoffs, including the benchmark-specific Hybrid constructors. None retains the shared state channel; B-only receives no A experience.

Our empirical contribution evaluates **accepted experience as a memory update**, against both its predecessor and independence. B-only/U/C separates correction gains, baseline restoration, and success beyond both references. Repair-by-representation comparisons show why **correction sensitivity need not rank corrected-memory quality**: Full's larger gap over Skill mostly reflects weaker U. ThinkingBox's workflow gains and both APEX regimes' weaker aggregate transfer motivate an applicability hypothesis. Tokens and calls distinguish reading savings from execution savings.

2 Related Work

From repairing a task to transferring its correction. Self-Refine and Reflexion improve attempts through feedback (Madaan et al., 2023; Shinn et al., 2023); human–AI collaboration distinguishes correction, remembering, and adaptation (Amershi et al., 2019). We compare pre/post-repair experience on a distinct B, making A's acceptance the starting point rather than the transfer outcome.

Experience reuse and representation. ExpeL, CLIN, and Agent Workflow Memory extract knowledge and workflows (Zhao et al., 2024; Majumder et al., 2024; Wang et al., 2025); Trace2Skill studies skill reuse (Ni et al., 2026). ReasoningBank learns from successes and failures (Ouyang et al., 2026). *Break It Down, Pass It On* studies granularity, text/code formats, and negative transfer (Feng et al., 2026); SkillsBench measures gains and regressions (Li et al., 2026a). Our axis is *revision of the same failed episode*: fixed-B, pre/post-repair comparisons measure the update's incremental value; B-only separates restoration from success absent under both references.

Content, state, and the work of using memory. Prompt compression (Jiang et al., 2023), masking versus summaries (Lindenbauer et al., 2025), repository context (Gloaguen et al., 2026), and cross-session memory-guided action (He et al., 2026) connect retention to performance and cost. We compare accepted-segment content, state/text handoffs, and token versus action savings (Appendix G).

3 DESIGN: FROM AN ACCEPTED REPAIR TO A NEW TASK

Each regime d (ThinkingBox/APEX-state/APEX-text) has $N = 100$ source–target pairs (A_i, B_i), indexed by $i = 1, \ldots, N$. For branch $s \in \{U, C\}$, z^s_{Ai} records A's history and saved state. The binary source acceptance function V_{Ai} (1: accepted) selects $V_{Ai}(z^U_{Ai}) = 0$, $V_{Ai}(z^C_{Ai}) = 1$, with C the first accepted repair within ten feedback–repair rounds.

Two channels of experience. For format r (Table 1), initialization $\mathcal{J}$ maps B's original environment e^0_{Bi} and A's record to starting environment $\widetilde{e}$ and shared state-description text v; ϕ constructs episode text m. Outputs suppress d:

$$(\widetilde{e}^{s,r}_{Bi}, v^{s,r}_i) = \mathcal{J}_{d,r}(e^0_{Bi}, z^s_{Ai}), \quad m^{s,r}_i = \phi_{d,r}(z^s_{Ai}), \tag{1}$$

$$\tau^{s,r}_{Bi} \sim \text{Rollout}(\pi_\theta, x_{Bi}, \widetilde{e}^{s,r}_{Bi}, v^{s,r}_i \oplus m^{s,r}_i). \tag{2}$$

Rollout samples ($\sim$) trajectory τ—actions, observations, and outputs—using policy π_θ, fixed model parameters θ, B's request x_{Bi}, and text concatenation $\oplus$. Text construction sees only A; v can be empty. Only physical inheritance changes $\widetilde{e}$ from e^0.

ThinkingBox supplies terminal-state text to B's independent sandbox. In APEX, W is the original world, I_A/I_B the source/target input bundles, and Δ^s_A saved source-branch changes. Arrows mean sequential loading/application; i is omitted:

$$\text{APEX-state} : W \to I_A \to \Delta^s_A \to I_B, \qquad \text{APEX-text} : W \to I_B.$$

APEX-state carries 649 filesystem entries and two deletions across 200 A deltas, with no identified database payload. Both regimes share target identities and rubrics.

3.1 WHAT EXACTLY IS HANDED OVER?

None/Full/Skill/Hybrid use both U and C; Summary/Execution use C alone. With B-only, Table 1's six formats give eleven conditions. None retains terminal-state text in ThinkingBox/APEX-text or restored state in APEX-state; it is not B-only. Text Full/Skill/Hybrid share None's branch prefix. Summary/Execution share initialization and summary, adding only the accepted segment, which may depend on earlier A work. Full–Execution replaces the whole text handoff.

Table 1: A-derived episode content in each handoff. Every format also retains its regime's shared state channel. No constructor sees B's request or score.

Format	Episode content given to B
None	Shared state channel only; no additional episode text.
Full	Chronological actions, observations, and any failure/feedback/repair, subject to trace limits.
Skill	Abstract procedures, constraints, checks, and error–correction rules.
Hybrid	ThinkingBox: history summary plus four recent raw assistant segments. APEX: the same Skill plus grounded facts and provenance.
Summary	C-only; at most 350 requested words on accepted results, entities, and decisive actions; no earlier failures/feedback.
Execution (Exec)	Identical Summary plus the final accepted segment's actions, tool calls, and observations; not the entire repair history.

Formats vary in selection, order, abstraction, and length, not a single compression level. U can mark uncertainty and C user validation. C–U changes a correction-associated package; token count and authority cues are not independently manipulated.

3.2 QUESTIONS AND CONTRASTS

RQ1: Does repairing A make its experience more useful to B? H1a/H1b predict that C-Full/Skill/Hybrid outperform their U versions/B-only, respectively. **RQ2: What should be retained after acceptance?** H2a predicts Execution exceeds Summary; H2b predicts C-Full exceeds Execution. **RQ3: Does a smaller successful-experience handoff reduce B's work?** Relative to C-Full, H3a/H3b predict that Execution reduces cumulative input tokens/model calls, respectively.

B's binary acceptance function V_{Bi} evaluates its trajectory and outputs: $Y_{di}^{s,r} = V_{Bi}(\tau_{Bi}^{s,r})$; Y_{di}^{0} is B-only acceptance. Mean differences estimate correction (corr: C–U) and baseline-relative (base: C–B-only) gains:

$$\widehat{\Delta}_{d,r}^{\text{corr}} = \frac{1}{N}\sum_{i=1}^{N}(Y_{di}^{C,r} - Y_{di}^{U,r}), \qquad \widehat{\Delta}_{d,r}^{\text{base}} = \frac{1}{N}\sum_{i=1}^{N}(Y_{di}^{C,r} - Y_{di}^{0}). \tag{3}$$

4 Evaluation and Statistical Analysis

ThinkingBox has 65 insurance/35 consulting pairs in 15 domain–family groups. APEX has 48 banking/23 law/29 consulting pairs in 16 worlds: 42 same-family/26 shared-procedure/20 shared-evidence/12 context-only, annotated from full requests. Both APEX regimes share identities and labels: **200 distinct pairs, not 300 independent pairs** overall.

ThinkingBox uses executable assertions; APEX judges B's answer/artifacts against its original rubric with native visual evidence. Text artifacts reconstruct B's original inputs plus its delta, not A. Each condition retains 100 targets, including budget failures. State/text rubric comparisons align partial-artifact scoring; original judgments and technical provenance remain separately auditable (Appendix F).

Acceptance uses two-sided exact McNemar tests, with Holm families of nine H1a, nine H1b, and six H2 contrasts, plus global-24 sensitivity. We report raw p, family-adjusted p_H, and global-adjusted p_{24}; pp denotes percentage points. Pointwise paired/group bootstrap intervals retain pairing and 15/16 groups. Exhaustive group sign flips use the same families, requiring independent groups and within-group joint exchangeability under the null, not causal randomization. Exploratory format tests form separate families of 36 within-branch and 18 correction-gap contrasts. Rubric means weight tasks equally. Analyses are retrospective, not preregistered.

APEX uses `deepseek-flash` with thinking, 100 steps, and 7,200 seconds; ThinkingBox defaults to `deepseek-v4-flash`, high reasoning, 400 interactions, and 1,200 seconds. Regime-specific settings are fixed (Appendix A).

5 RQ1: When Does Corrected Experience Transfer?

Table 2: Absolute target acceptance (%). Every cell has 100 targets, including budget failures. State/Text are the two APEX regimes.

Condition	ThinkingBox	State	Text
B-only	42	44	47
U-None / C-None	37 / 36	45 / 44	45 / 47
U-Full / C-Full	23 / 67	42 / 41	45 / 45
U-Skill / C-Skill	35 / 64	39 / 42	44 / 45
U-Hybrid / C-Hybrid	28 / 60	41 / 45	48 / 40
Summary / Execution	56 / 67	40 / 44	40 / 52

5.1 Correction helps across representations in ThinkingBox, but not generally in APEX

An accepted repair is useful in one setting without being a general transfer signal. ThinkingBox C-Full accepts 67 of 100 targets, compared with 23 for U-Full and 42 for B-only. Repair therefore improves on both the failed memory and independent execution, rather than merely making a bad handoff less bad. Skill and Hybrid show the same ordering: their corrected versions accept 64 and 60 targets, above both their U versions (35 and 28) and B-only. In contrast, APEX-state C-Full accepts 41 targets against U-Full's 42 and B-only's 44; APEX-text C-Full and U-Full both accept 45, below B-only's 47. Figure 2 shows that neither APEX regime establishes a comparable aggregate advantage across the three representations. Source acceptance alone thus does not distinguish an update worth reusing from one that only solved A.

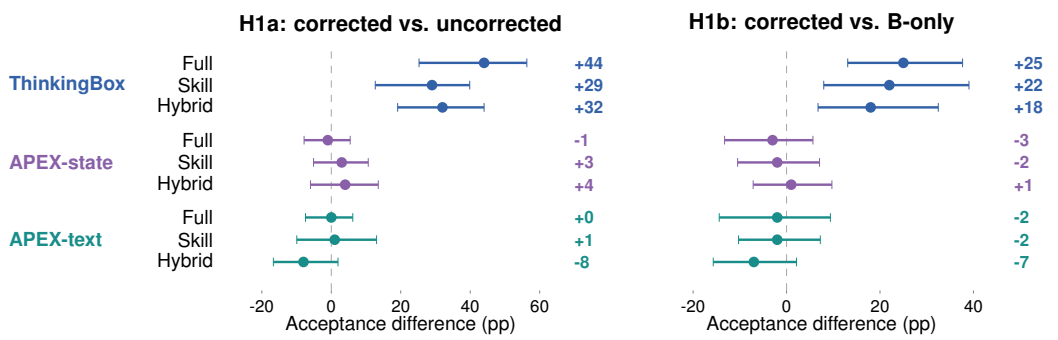


Figure 2: Dots/bars: paired acceptance differences/pointwise 95% cluster-bootstrap intervals; 100 targets/row. Numeric labels are estimates, not adjusted significance.

The ThinkingBox pattern is not confined to one representation or one family. For 42 targets, changing U to C turns failure into success under at least two of Full, Skill, and Hybrid; 18 improve under all three. The corresponding counts are only five and one in APEX-state, and six and one in APEX-text (Table 15). This asks whether the same targets benefit when the repaired episode is represented differently, not merely whether three aggregate means rise. ThinkingBox's overlap points to useful content surviving different representations. Its C–U advantages also remain positive after removing any single family, so no one family alone accounts for the observed direction.

5.2 A CORRECTION GAIN CAN RESTORE A SUCCESS THAT MEMORY DISPLACED

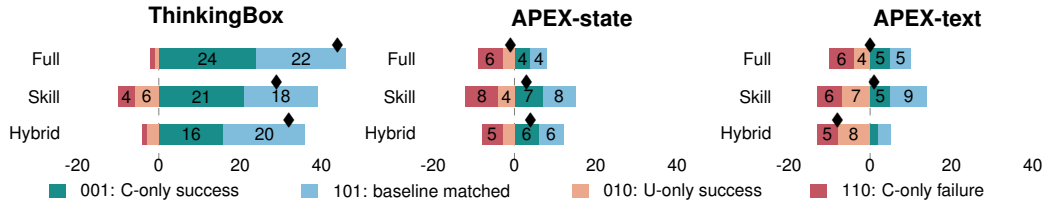


Figure 3: B-only/U/C transitions per 100 targets, not individual causal effects. Bars distinguish baseline outcomes; diamonds show net C–U. Table 11 includes unchanged patterns.

Fix d, r and omit them below. Let n_{buc} count targets with B-only/U/C outcomes $(b, u, c) \in \{0, 1\}^3$, where 1 denotes acceptance. Net transfer decomposes as

$$N\widehat{\Delta}^{\text{corr}} = \underbrace{n_{001}}_{\text{C-only success}} + \underbrace{n_{101}}_{\text{baseline matched}} - \underbrace{n_{010}}_{\text{U-only success}} - \underbrace{n_{110}}_{\text{C-only failure}} .$$

ThinkingBox Full produces 46 upward transitions from U to C and two downward transitions, giving its 44 pp net gain. But 22 of those 46 improvements occur where B-only already succeeds: U loses that success and C restores it. The other 24 occur where both B-only and U fail. These are different reasons to value correction. Restoration justifies replacing a misleading memory already in use; success beyond both references supports introducing the corrected experience instead of starting independently. Reporting only C–U merges these decisions and would count nearly half of Full's upward transitions as if they were successes unavailable without memory.

Conversely, a zero mean need not mean that correction changes nothing. APEX-text Full has ten U-failure/C-success transitions and ten U-success/C-failure transitions: the acceptance rate stays at 45%, while twenty targets change outcome. APEX-state Full similarly has eight gains and nine losses. There is room to improve—text U-Full fails on 55 targets—but the successes recovered by C are offset by successes lost elsewhere. Thus the APEX result is not simply saturation or complete irrelevance of the handoff. It is an unfavorable balance between helping some targets and preserving performance on others. The transition counts are observed paired outcomes, not individual causal effects, but they identify what an aggregate transfer score conceals.

5.3 THE REPRESENTATION THAT IMPROVES MOST IS NOT NECESSARILY THE BEST TO REUSE

Full's larger correction gain mostly reflects its weaker starting point. In ThinkingBox, Full moves from 23% under U to 67% under C; Skill moves from 35% to 64%. Full's correction gain is therefore fifteen points larger, although its corrected acceptance is only three points higher. Twelve of the fifteen points arise because U-Full performs worse than U-Skill. The three-point C-Full advantage is not established by the adjusted representation tests (Appendix B). This separates two questions: how strongly a representation responds to repair, and which corrected representation should be supplied to B. Ranking the first quantity as though it answered the second would overstate the case for retaining the full history.

None shows that episode content matters; it does not tell us how much detail to keep. ThinkingBox C-None retains the terminal-state description and accepts 36 targets. C-Full, C-Skill, and C-Hybrid accept 67, 64, and 60. Their shared advantage over this minimal channel means the observed benefit is not reproduced by reporting the source's terminal state alone. Yet the three richer formats need not preserve the same information or solve the same targets. APEX-text makes this distinction concrete: C-Full and C-Skill each accept 45 targets, but disagree on 24, with twelve successes exclusive to each. Equal acceptance therefore does not make compression behaviorally neutral; the abstract representation exchanges some successes for others rather than simply retaining a smaller equivalent memory.

Hybrid tests whether restoring detail to an abstraction improves that tradeoff. APEX Hybrid adds grounded facts to the same Skill, but corrected acceptance moves from 42% to 45% with state inheritance and from 45% to 40% without it. Neither adjusted difference is established. The results offer no general ordering in which adding facts is always better, just as they do not establish Skill's noninferiority to Full. Full retains context and competing attempts; Skill foregrounds procedures; Hybrid restores selected particulars. Their practical difference is which evidence B can use to decide whether a source lesson applies, not merely how many tokens are removed. The behavioral cases below examine that decision directly.

5.4 BEHAVIOR DISTINGUISHES REUSABLE OBLIGATIONS FROM SOURCE-SPECIFIC CHOICES

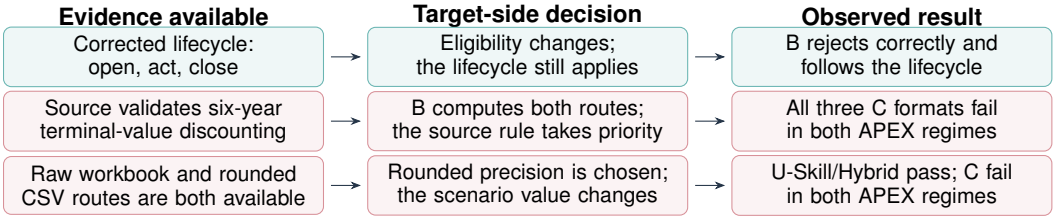


Figure 4: Selected action chains, not frequencies: lifecycle (ThinkingBox), discounting/precision (APEX). Both APEX regimes show the latter; displayed decisions come from text trajectories.

A workflow obligation can remain useful even when the answer changes. ThinkingBox's lifecycle rule requires opening a ticket, acting, and then closing it. In `lif_017`→`lif_015`, B's eligibility differs from A's, so copying A's business decision is inappropriate. B-only and U-Skill already reject B correctly, but create an already-solved ticket; C-Skill rejects correctly and follows the required lifecycle. Correction contributes how to carry out the decision, not which eligibility answer to copy. The broader validator pattern is consistent with this example: lifecycle violations fall from 55 to two under Full, 40 to seven under Skill, and 48 to three under Hybrid. Of the 54 flags removed under Full, 37 targets pass and 17 still fail. A reusable procedure can therefore explain an important behavioral improvement without accounting for every remaining task requirement.

An accepted local choice can instead override the new request. In APEX pair `ad52b8019140`, B explicitly requests mid-year discounting. Text C-Skill describes full-year terminal discounting as user-approved. B computes both the six-year and 5.5-year alternatives, yet selects the former, producing 40.45/44.83 rather than 41.62/46.22. The missing capability is not calculating the alternative: it appears in the trace. The failure is selecting a source-approved convention despite the target's different requirement. B-only and all U formats pass in both APEX regimes, while all C

formats receive zero credit. Its presence with and without inherited artifacts shows that this pattern is not confined to reconstructing missing source files.

The contrast is about *scope*, not a rule that detailed memories are harmful. In Helios, correction supplies the useful scope restriction: text U-Hybrid combines 4,900 SKU rows, whereas C-Hybrid uses B's specified 2,500-row Rebuilt input and recovers the correct result. Both APEX regimes show B-only pass, U-Hybrid fail, C-Hybrid pass, making this a concrete instance of baseline restoration. Correction can thus narrow attention to the right input or elevate the wrong convention. Source approval alone does not tell B which of these it is doing; the target requirement must resolve the choice.

These traces motivate preserving an obligation together with the conditions that justify it. They do not establish a causal effect of an approval marker or a population frequency of this mechanism. Precision and workbook-lineage cases also require care: Lumea's rounded-input choice changes the result, whereas 3M's target does not specify which workbook version to use. A lower fixed-rubric score in the latter is not sufficient evidence of harmful reuse. Appendix D retains these selected, non-blind audits; Appendix C.2 reports domain and state/text comparisons. The main distinction is supported by target-specific action evidence: *carrying over a procedure that still applies is different from carrying over a decision whose premises have changed.*

6 RQ2 AND RQ3: ACTION DETAIL, ACCEPTANCE, AND TARGET-SIDE COST

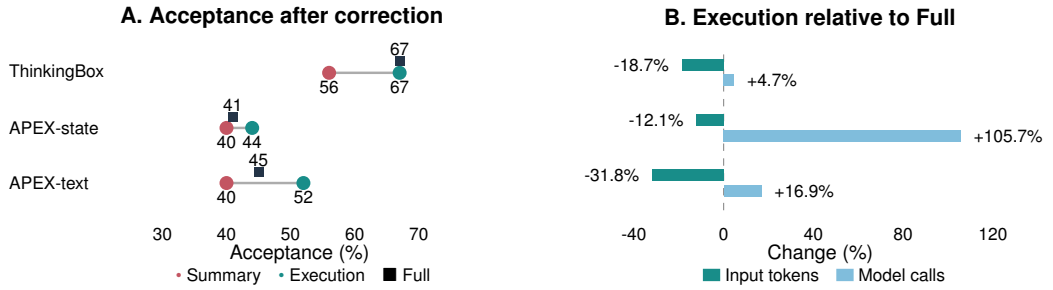


Figure 5: Acceptance per 100 targets (left); mean Full-to-Execution input/call changes (right). Tables 21–22 report exact means and tests.

6.1 ACTION DETAIL IMPROVES ON A SUMMARY, NOT NECESSARILY ON STARTING FRESH

Summary and Execution share the same accepted-source summary and initialization; Execution additionally supplies the accepted segment's actions, tool calls, and observations. Their contrast asks whether showing how A was completed adds value beyond describing its result. In APEX-text, Execution accepts 52 targets against Summary's 40: fourteen targets change from failure to success and two change in the opposite direction. ThinkingBox likewise rises from 56 to 67, whereas APEX-state rises from 40 to 44 (Figure 5). The common positive direction favors retaining operational evidence, but its magnitude and statistical support differ across settings.

Text Execution's twelve-point advantage over Summary is only five points over B-only's 47%, with a world interval crossing zero for the latter. Improving a summary-based handoff therefore does not establish superiority over independent execution. Text H2a survives target-level H2-family adjustment ($p_H = 0.0251$), but not global-24 or group adjustment; the other regimes do not pass H2-family adjustment. Full also establishes no advantage over Execution. The clearest observed benefit is thus adding action evidence to a summary, not retaining the entire repair history.

6.2 NAMING A REQUIREMENT IS DIFFERENT FROM CARRYING IT OUT

Two ThinkingBox gains make the added evidence interpretable. Summary already instructs B to set the ticket type to `task`, yet B fails to set the field; with Execution, it does. The requirement was available in both handoffs, so these cases are not explained by its absence from the summary. Execution additionally shows concrete operations and observations. The contrast suggests that a

handoff can fail between recognizing a requirement and implementing it, making operational examples useful even when the summary names the right rule. It does not isolate which part of the added segment produced the improvement.

6.3 A SMALLER HANDOFF REDUCES CONTEXT READING BUT CAN REQUIRE MORE EXECUTION

Relative to Full, Execution reads less cumulative input in all three regimes, yet makes more model calls. APEX-text illustrates the tradeoff: mean input falls from 3.576M to 2.438M tokens (31.8%), while calls rise from 30.01 to 35.08 (16.9%) and tool calls from 38.24 to 44.40. The explanation at the accounting level is that input per call falls from 119.2k to 69.5k: each interaction reads less context, enough to outweigh the additional interactions. H3a and H3b therefore separate. Reducing repeated context consumption is not the same achievement as reducing the work needed to complete B.

APEX-state approximately doubles calls under Execution (20.04 to 41.22), limiting its input reduction to 12.1%; ThinkingBox's calls rise only 4.7% while input falls 18.7%. More work remains downstream, although extra calls need not all be reconstruction. Which representation is preferable depends on the constraint: in text APEX, 32 accepted Execution runs versus eleven Full runs fit within two million input tokens, but only ten versus eighteen fit within twenty model calls. These are completed-run counts, not early-stopping experiments. Execution also uses 12.2% more input than Summary. Memory selection must therefore consider success, context consumption, and interaction demand jointly. Appendix E gives uncertainty and accounting boundaries; B-stage token reductions are not billing or whole-pipeline savings.

7 DISCUSSION: WHAT DOES REPAIR ADD TO REUSABLE EXPERIENCE?

The same-episode comparison evaluates a revision, not a category of memories. Learning from successful and failed experience is already central to agent memory (Ouyang et al., 2026; Feng et al., 2026). Our comparison asks what is gained by repairing *a particular failed episode*. Holding its provenance and target fixed reveals a distinction that source acceptance alone cannot resolve: a revision may improve on its predecessor without being preferable to independent execution. C–U addresses the first decision; C–B-only addresses the second. A system selecting memory updates needs both.

ThinkingBox Full moves from 23% to 67%, whereas Skill moves from 35% to 64%. Their 44- and 29-point correction gains would make Full look substantially better if only C–U were considered. Yet twelve of the fifteen extra points come from Full's worse uncorrected version; the corrected versions differ by only three points. A larger response to repair can reflect a weaker predecessor, rather than a better final representation.

Repair can recover performance that memory itself displaced. Full's 46 upward transitions contain two different outcomes. On 22 targets, B-only already succeeds, U fails, and C succeeds: repair recovers the result available without source experience. On the other 24, both B-only and U fail before C succeeds. Both count as gains over U, but only the second pattern records success beyond both references. Restoration remains valuable when misleading memory is already in use; it is not the same justification for introducing memory in the first place. For memory updates, the useful question is consequently not just how many failures were fixed, but which previous successes were retained and which successes became available beyond a fresh start.

A repaired solution mixes general obligations with local decisions. Behavior traces suggest why source improvement and transfer value diverge. In ThinkingBox, a changed eligibility decision still requires the same open–act–close workflow: the obligation remains applicable when the answer changes. In APEX, the source-approved full-year convention conflicts with B's explicit mid-year requirement. B computes both alternatives but selects the source convention. The problem is not a missing procedure; it is using a locally valid choice outside the conditions that justified it. Both APEX regimes show this case, while neither establishes ThinkingBox-like aggregate correction gains.

This distinction suggests a concrete memory-design objective: preserve the useful correction without turning every accepted source decision into a default rule. An entry would retain what was changed, the requirement that justified the change, and the inputs or artifacts on which it depended. The target agent would then check whether those conditions still hold. For example, a ticket lifecycle can remain relevant across customers, whereas a workbook version or discounting convention must be checked against the new request. Acceptance supplies evidence about what worked on A; B's requirements determine what can be reused.

Representation and state change the work required to make that decision. Full can supply context for reconstructing a rule's scope; Skill makes the rule easier to find but can omit that context. Hybrid adds detail, which helps only if it supports the right target-side decision. Execution can show how to carry out an action that Summary merely names. The choice is not simply more versus less memory (Lindenbauer et al., 2025). APEX-state makes source files available, whereas APEX-text can require rebuilding them; neither establishes that their assumptions fit B. In Planet Fitness, reconstruction changes performance without changing C–U. A useful handoff must support both selecting and applying appropriate knowledge. Its cost includes the resulting target work: our smaller Execution handoffs reduce input but increase calls.

8 Scope and Open Questions

The next question is whether explicitly representing applicability improves transfer. A focused test would compare the same C-Skill rules unchanged, with a generic reminder to check B, and with source-derived conditions specifying when each rule applies, holding the model and budget fixed. The reminder separates the value of those conditions from simply encouraging more caution. On held-out targets, with constructors restricted to A, evaluation should measure whether convention conflicts decrease *while useful workflow behavior is retained*. Generic-checklist and approval-marker controls would distinguish shared norms, task-specific lessons, and source authority.

The present evidence concerns complete repair packages, failed-but-repairable sources, and one execution per cell. Group-level inference addresses task dependence, not repeat-run reliability; repetitions and blinded artifact adjudication would distinguish persistent gains from execution and scoring variation. Crossing state with identical text would isolate inheritance, and actual database changes would extend the file-state evidence. A longer-term question is whether successive repairs accumulate useful knowledge or obsolete commitments. Studying target-aware reuse with natural human feedback, including supervision, memory construction, and reconstruction costs, would connect the current one-handoff comparison to a continuing memory system.

9 Conclusion

Repairing a failed episode and obtaining useful experience from it are distinct achievements. The same-episode design shows why: a large correction gain may reflect a poor uncorrected memory, and an improved target outcome may restore performance already available without memory. ThinkingBox also shows success beyond both references, while the two APEX regimes do not establish comparable aggregate benefits. Repair matters, but source acceptance alone does not say when its lessons should be reused.

The resulting principle is to evaluate a memory revision against both its predecessor and a fresh start, and to preserve the conditions under which its advice was valid. This shifts the design question from storing an accepted solution to deciding which parts should guide the next task. After the fix, useful experience must help B choose what still applies, not merely reproduce what worked on A.

Reproducibility Statement

The analysis uses 3,300 outcome-and-cost records, 1,100 canonical APEX-text reviews, and 20 separate file-state alignment reviews, with paired-analysis scripts, source identifiers, cohort construction, settings, prompts, and editable figures. Every benchmark–condition retains 100 targets. Transition counts and cross-representation consistency are regenerated from the same frozen records.

Ethics and Data Use

Feedback is supplied by simulated users; the study involves no human participants. Benchmark materials are used for evaluation, not parameter training. Access and redistribution follow their respective conditions. The statistical package excludes credentials and restricted benchmark attachments.

AI Use Statement

Generative AI was used in the experimental pipeline for task pairing, agent execution, simulated user feedback, constructing source-derived experience handoffs, and APEX artifact judging. Model configurations, prompts, and evaluation procedures are documented in Appendices A and F. OpenAI Codex also assisted with refining research questions, hypotheses, and conceptual framing; implementing and debugging experimental and statistical-analysis code; literature search and synthesis; qualitative trajectory and artifact analysis; interpreting results; and drafting, revising, translating, and formatting the manuscript and figures. Quantitative results are computed by scripts from saved execution and judgment records. Verification includes script-based consistency checks, tests of statistical-analysis routines, and inspection of saved trajectories, artifacts, and cited sources, with judging audits and sensitivity analyses described in Appendix F. The authors retain responsibility for the research design, interpretation, and final content, including AI-assisted text, code, and artifacts.

References

Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. Guidelines for human-AI interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2019. doi: 10.1145/3290605.3300233.

Yiyang Feng, Biddut Sarker Bijoy, Niranjan Balasubramanian, and Jiawei Zhou. Break it down, pass it on: Cross-task skill transfer in LLM agents. *arXiv preprint arXiv:2608.20274*, 2026. URL https://arxiv.org/abs/2608.20274.

Thibaud Gloaguen, Niels Mündler, Mark Müller, Veselin Raychev, and Martin Vechev. Evaluating AGENTS.md: Are repository-level context files helpful for coding agents? *arXiv preprint arXiv:2602.11988*, 2026. URL https://arxiv.org/abs/2602.11988.

Zexue He, Yu Wang, Churan Zhi, Yuanzhe Hu, Tzu-Ping Chen, Lang Yin, Ze Chen, Tong Arthur Wu, Siru Ouyang, Zihan Wang, Jiaxin Pei, Julian McAuley, Yejin Choi, and Alex Pentland. MemoryArena: Benchmarking agent memory in interdependent multi-session agentic tasks. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.16313v2.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=IkmD3fKBPQ.

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*, pp. 13358–13376, 2023. doi: 10.18653/v1/2023.emnlp-main.825.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. *arXiv preprint arXiv:2505.06120*, 2025. URL https://arxiv.org/abs/2505.06120.

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, Kyoung Whan Choe, Jiankai Sun, Shuyi Wang, et al. SkillsBench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670v4*, 2026a. URL https://arxiv.org/abs/2602.12670v4.

Zhuochun Li, Youngmin Ko, Ali Keramati, Nicola Ferri, Susana Palmaz Lopez Pelaez, Liang-Chun Tsai, Calvin Wang, Mirco Milletari, Tuhin Kundu, Vadim Smolyakov, Kjartan Olafsson, and Tommy Guy. One success isn't reliability: Thinkingbox, a sandbox and benchmark for agents in stateful business workflows. *arXiv preprint arXiv:2608.19741*, 2026b. URL https://arxiv.org/abs/2608.19741.

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.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594, 2023. doi: 10.52202/075280-2019.

Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. CLIN: A continually learning language agent for rapid task adaptation and generalization. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=xS6zx1aBI9.

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. Trace2Skill: Distill trajectory-local lessons into transferable agent skills. *arXiv preprint arXiv:2603.25158*, 2026. URL https://arxiv.org/abs/2603.25158.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. ReasoningBank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jL7fwchScm.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652, 2023. doi: 10.52202/075280-0377.

Bertie Vidgen, Austin Mann, Abby Fennelly, John Wright Stanly, Lucas Rothman, Marco Burstein, Julien Benchek, David Ostrofsky, Anirudh Ravichandran, Debnil Sur, Neel Venugopal, Alannah Hsia, Isaac Robinson, Calix Huang, Olivia Varones, Daniyal Khan, Michael Haines, Austin Bridges, Jesse Boyle, Koby Twist, Zach Richards, Chirag Mahapatra, Brendan Foody, and Osvald Nitski. APEX-Agents. *arXiv preprint arXiv:2601.14242*, 2026. URL https://arxiv.org/abs/2601.14242.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63897–63911, 2025. URL https://proceedings.mlr.press/v267/wang25bx.html.

Gregory Yauney, Shahzaib Saqib Warraich, and Swabha Swayamdipta. How reliable is language model micro-benchmarking? In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2e2960f2fe9e981f33f51c78656e3ca2-Abstract-Conference.html.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936.

Xu Zou and Jie Tang. Trace as state: Reasoning traces as conditional states for long-context transformers. *arXiv preprint arXiv:2609.02702*, 2026. URL https://arxiv.org/abs/2609.02702.

A PROTOCOL, COHORTS, AND EXECUTION SETTINGS

Equations 1–2 define the initialization and textual channels. Table 3 specifies their implementation in each regime, and Table 4 distinguishes the quantities held fixed by each contrast. Source records and constructors are regime-specific; memory construction has no access to B's request.

Table 3: State and text channels. W is the original APEX world, I_A/I_B the task inputs, and Δ_A^s branch-specific source changes. B-only excludes A in every regime.

Regime	B's starting environment	A-derived information
ThinkingBox	Independent standard B sandbox	Terminal-state description and the specified textual representation; no live A database restoration
APEX-state	$W \to I_A \to \Delta_A^s \to I_B$	Source environment state plus the specified text; None supplies the state channel alone
APEX-text	$W \to I_B$	Text alone; None supplies terminal observations, with the same branch prefix in Full/Skill/Hybrid

Table 4: Experimental contrasts and their interpretation.

Contrast	Held fixed	What the contrast measures
C–U, same representation	Target and evaluation	The complete correction-associated experience package
C–B-only	Target and evaluation	Benefit beyond independent target execution
Execution–Summary	Exact summary, framing, starting state	Adding the accepted segment, including its content and length
Full–Execution	Target and evaluation	Alternative whole handoffs, not history alone
APEX-text–APEX-state	Pair identities and target rubric	Implemented handoff regimes, with state and textual-construction differences

ThinkingBox selection. The source benchmark is ThinkingBox (Li et al., 2026b). Task identifiers are grouped by domain and workflow-family prefix, shuffled with seed 260914, and paired cyclically within each family before formal A outcomes are observed. Screening includes 79 insurance and 81 consulting candidates: 19/46 initially pass, leaving 60/35 failures. Five pilot failures supplement these 95. Eight sources remain unaccepted after ten feedback rounds and are replaced by eight unused natural pilot failures. The final cohort contains 87 formal-screen and 13 pilot sources, not a held-out cohort. Its accepted repairs occur at rounds one through four for 78/13/6/3 sources. The full task mapping is included in `data/cohort_construction.json`.

ThinkingBox retries restore A's canonical sandbox because erroneous writes to append-only audit records cannot always be undone; visible failure, feedback, and repair are retained in the episode. Each B starts in its own standard sandbox. This differs from physically carrying A's database forward and is why the state channel is textual in Equations 1–2. Feedback privately uses the executed trace and validator differences, asks for the earliest causal mistake, and must translate it into user-facing language without exposing reference actions, assertions, or scores. It is simulated feedback with privileged diagnostic access, not a study of unaided human review.

Shared APEX cohort. Both regimes use the APEX-Agents benchmark (Vidgen et al., 2026). The pairing stage assigns a within-world one-to-one mapping with no self-pairs across 480 tasks in 33 worlds. The pairer sees requests and delivery modes, not outcomes, gold answers, or rubric scores; malformed mappings are completed by constrained assignment using suggested pairs and lexical similarity. These prospective pairings are distinct from the retrospective relationship annotations in Appendix C.2. Five pilot A attempts are reused. Of the other 475, 462 complete and are reviewed: 227 initially pass and 235 are correction candidates. The selection gate records 198 accepted repairs plus three accepted pilot repairs, giving 201 eligible pairs. Both APEX regimes use the same first 100 pairs in frozen manifest order; selection is not optimized on B results and yields 16 worlds. All

C branches use their first saved accepted repair. The cohort estimates utility conditional on failure, repairability, and selection order, not population performance across all APEX-state tasks.

Source feedback and compression. APEX-state review inspects the task, rubric, hidden reference when available, final answer, tool trace, and changed artifacts. It produces criterion judgments and natural-language corrective feedback; hidden reference values are not intentionally passed through the feedback prompt. Memory generation receives the source request, visible user turns, final answer, tool digest, and changed artifacts, not B. Its prompt distinguishes user-validated corrections from uncertain unresolved tactics. Skill stores procedural knowledge; Hybrid appends concrete facts and provenance to the same Skill. ThinkingBox compression produces a state-and-action summary (up to 220 words), procedural skill (450 words), and chronological summary (900 words); Hybrid combines the latter with four recent assistant-turn segments. Full follows per-message and total trace truncation. These are requested budgets, not guaranteed token counts.

Table 5: Execution settings. Output budgets are tokens per call. The package contains full prompt constructors and saved configuration fields.

Component	ThinkingBox	APEX-state
Acting model	Environment-resolved DeepSeek alias; adapter default `deepseek-v4-flash`	DeepSeek V4.1 Flash (multimodal), API `deepseek-flash`
Agent sampling	Thinking enabled; explicit `reasoning_effort=high`; temperature omitted	Thinking enabled; no explicit reasoning effort; temperature 0.2; no supplied seed
Agent budgets	32,768 output; 400 interactions; 30 user turns; 1,200 seconds	8,192 output; 100 steps; 7,200 seconds
Runtime user	Temperature 0.2; 4,096 output; cannot terminate agent conversation	No separate conversational user during B
Correction review	Temperature 0.2; 2,048 output; trace and state differences	Temperature 0; 8,192 output; artifacts and criteria
Memory compressor	Temperature 0; 8,192 output; thinking disabled	`deepseek-v4-pro`; temperature 0; 8,192 output
B acceptance	Executable task assertions; configured legacy judge where applicable	Owner review; delivery integrity and all criteria satisfied
Version record	Resolved alias and provider revision unavailable	All 1,100 target summaries record `deepseek-flash`

The prompts are exported from the implementation: `REVIEWER_SYSTEM`, the U/C memory constructors, terminal-state extraction, APEX-state `SYSTEM_PROMPT`, review, compression, and pairing, plus the runtime user's instructions. The retained ThinkingBox terminal-field rerun container records `MODEL_NAME=deepseek-v4-flash`; earlier deleted containers' resolved settings cannot be inferred from it. The package excludes credentials and benchmark attachments. Re-execution requires authorized materials, sandbox services, and runner dependencies. An immutable historical API model cannot be reconstructed from an unrecorded revision.

Successful-experience handoffs. The Success-summary constructors for both benchmarks are target-blind and request at most 350 words covering observable results, entities, and decisive actions. Earlier failures and user corrections are excluded from this summary input. Execution adds the final accepted source segment to the identical summary. In ThinkingBox the accepted attempt starts from A's restored standard sandbox; in APEX-state it can be a final repair over artifacts already modified by preceding attempts. Thus an accepted segment is not necessarily an end-to-end replay from the original environment. APEX-state retains up to 16,000 characters per message and 140,000 characters overall. Both added APEX-state conditions restore identical C-state changes before B; ThinkingBox uses B's standard sandbox. Full reuses C-Full's original prompt, not a nested extension of Execution.

Text-only initialization and state inventory. APEX-text starts B from the original world and B inputs, supplying A-derived text without restoring A's inputs or source delta. Its acting model is `deepseek-flash`, with thinking enabled, 100 steps, 8,192 output tokens, and a 7,200-second task timeout. It uses the same 100 pair identities and target rubrics as APEX-state. Terminal observations

and Full serialization differ across implementations, so the comparison is between implemented handoff regimes, not an isolated state intervention.

The inspected APEX-state payloads contain 649 filesystem files and two deletions across 200 unique A deltas; A inputs contain 45 filesystem files and one `.DS_Store`. No `.apps_data`, SQLite header, or database filename is observed. The realized transfer is therefore file-state transfer; support for database restoration in the environment does not establish that database content was transferred in this cohort.

B Complete Results and Statistical Comparisons

Every regime–condition has 100 targets. ThinkingBox measures executable acceptance; APEX reports owner acceptance and the mean fraction of rubric criteria satisfied per task. APEX-state uses the partial-artifact alignment layer described in Appendix F; unfinished executions retain their original status and the full denominator.

Table 6: All eleven conditions in all three regimes. Each entry has denominator 100. A/Q denotes acceptance percentage / task-normalized rubric percentage.

Condition	ThinkingBox	APEX-state A/Q	APEX-text A/Q
B-only	42	44 / 55.54	47 / 57.40
U-None	37	45 / 54.79	45 / 53.50
U-Full	23	42 / 54.00	45 / 55.83
U-Skill	35	39 / 50.89	44 / 54.29
U-Hybrid	28	41 / 53.11	48 / 56.80
C-None	36	44 / 55.38	47 / 56.10
C-Full	67	41 / 49.10	45 / 53.51
C-Skill	64	42 / 52.56	45 / 54.65
C-Hybrid	60	45 / 54.41	40 / 48.53
Summary	56	40 / 49.65	40 / 49.92
Execution	67	44 / 52.33	52 / 58.88

B.1 Paired estimands and comparison families

For candidate and reference outcomes on the same target, G counts reference failures becoming successes and L the reverse. The acceptance difference is $(G - L)/N$, while $(G + L)/N$ measures outcome reversal. Exact two-sided McNemar tests use the discordant pairs. Paired bootstrap intervals resample targets together with all their conditions; cluster intervals resample the 15 ThinkingBox domain–family groups or 16 APEX worlds. The archived ThinkingBox/APEX-state intervals use 20,000 resamples; text and matched-regime analyses use 10,000. All are pointwise, not simultaneous intervals.

The retrospective manuscript-wide Holm families contain nine H1a contrasts (C–U), nine H1b contrasts (C–B-only), and six H2 contrasts (Execution–Summary and Full–Execution). Table 7 also reports global Holm over all 24 tests. All six ThinkingBox H1 contrasts survive the global adjustment. APEX-text Execution–Summary has family $p = 0.0251$ and global $p = 0.0753$: its significance depends on the comparison scope. Domain and relationship analyses are exploratory.

Table 7: All 24 paired acceptance contrasts. G/L counts gains/losses; differences and world/family-cluster 95% intervals are percentage points. TB/State/Text denote the three regimes; F/S/H in condition names denote Full/Skill/Hybrid, and standalone B/S/E denote B-only/Summary/Execution. p_H uses the relevant 9/9/6 family; p_{24} uses all contrasts.

Regime	Cmp.	G/L	Gap	Cluster CI	Raw p	p_H	p_{24}
TB	C-F / U-F	46/2	+44	[25.3, 56.3]	< 0.0001	< 0.0001	< 0.0001
TB	C-S / U-S	39/10	+29	[12.7, 39.8]	< 0.0001	0.0003	0.0008
TB	C-H / U-H	36/4	+32	[19.1, 44.0]	< 0.0001	< 0.0001	< 0.0001
TB	C-F / B	31/6	+25	[13.1, 37.7]	< 0.0001	0.0004	0.0009
TB	C-S / B	30/8	+22	[8.0, 39.0]	0.0005	0.0038	0.0094

Regime	Cmp.	G/L	Gap	Cluster CI	Raw p	p_H	p_{24}
TB	C-H / B	24/6	+18	[6.7, 32.5]	0.0014	0.0100	0.0272
TB	E / S	15/4	+11	[1.4, 18.9]	0.0192	0.0961	0.3266
TB	C-F / E	7/7	+0	[-6.4, 6.1]	1.0000	1.0000	1.0000
State	C-F / U-F	8/9	-1	[-7.8, 5.4]	1.0000	1.0000	1.0000
State	C-S / U-S	15/12	+3	[-5.1, 10.6]	0.7011	1.0000	1.0000
State	C-H / U-H	12/8	+4	[-6.0, 13.5]	0.5034	1.0000	1.0000
State	C-F / B	10/13	-3	[-13.3, 5.6]	0.6776	1.0000	1.0000
State	C-S / B	12/14	-2	[-10.5, 7.1]	0.8450	1.0000	1.0000
State	C-H / B	11/10	+1	[-7.1, 9.7]	1.0000	1.0000	1.0000
State	E / S	12/8	+4	[-5.3, 13.3]	0.5034	1.0000	1.0000
State	C-F / E	11/14	-3	[-13.3, 6.5]	0.6900	1.0000	1.0000
Text	C-F / U-F	10/10	+0	[-7.4, 6.2]	1.0000	1.0000	1.0000
Text	C-S / U-S	14/13	+1	[-9.9, 13.0]	1.0000	1.0000	1.0000
Text	C-H / U-H	5/13	-8	[-16.7, 1.9]	0.0963	0.5775	1.0000
Text	C-F / B	11/13	-2	[-14.4, 9.4]	0.8388	1.0000	1.0000
Text	C-S / B	9/11	-2	[-10.3, 7.2]	0.8238	1.0000	1.0000
Text	C-H / B	7/14	-7	[-15.7, 2.1]	0.1892	1.0000	1.0000
Text	E / S	14/2	+12	[3.2, 22.5]	0.0042	0.0251	0.0753
Text	C-F / E	5/12	-7	[-19.8, 4.1]	0.1435	0.5739	1.0000

One execution per cell. For a fixed regime and target i, let c, c' denote two execution conditions, $Y_i(c)$ a binary acceptance result, and $p_i(c) = \Pr[Y_i(c) = 1]$ its repeated-run success probability. The quantity $w_i(c) = 2p_i(c)(1 - p_i(c))$ is the disagreement probability of two independent runs of the same condition. For independent draws $Y_i(c)$ and $Y_i(c')$,

$$\Pr[Y_i(c) \neq Y_i(c')] = (p_i(c) - p_i(c'))^2 + \tfrac{1}{2}\{w_i(c) + w_i(c')\}.$$

A single run cannot separate a condition's task-specific effect from execution variation. Resampling quantifies task-composition uncertainty; it does not estimate repeated-run reliability. Representations and shared APEX pairs are not independent replications.

B.2 Group-level test and recomputed interval sensitivity

For candidate c and reference c', define the target difference $D_i = Y_i(c) - Y_i(c')$ and group sum $S_g = \sum_{i \in g} D_i$, where g is a domain–family or world group. Let M be the number of groups (15 or 16). We enumerate all 2^M sign vectors with components $\epsilon_g \in \{-1, 1\}$ and report the fraction satisfying $|\sum_{g=1}^{M} \epsilon_g S_g| \geq |\sum_{g=1}^{M} S_g|$, retaining ties. This sign-flip sensitivity treats groups as independent and requires joint candidate/reference exchangeability within a group, or symmetry of group differences under the null. It is not a causal randomization test. The statistic retains target weighting rather than giving small worlds equal influence. With 15/16 groups, the support is discrete; grouping and multiplicity both reduce precision. No primary contrast survives its global-24 adjustment. Target-level significance and group-level sensitivity should not be conflated.

`scripts/analyze_representation_sensitivity.py` computes 20,000 paired and group resamples directly from the frozen cells with seed 20260925 (cohort offsets 0–2, matched-regime offset 3). A group draw retains every target in its sampled groups and divides by the resulting number of targets. All conditions are resampled jointly. The reproducible intervals below are an independently recomputed sensitivity, not the original archived random-number streams used in Figure 2. The script also recomputes Execution–Full resource intervals and five matched regime contrasts.

Table 8: Primary contrasts under group-level inference. CI: freshly recomputed pointwise cluster interval (pp); p_H uses the 9/9/6 families, p_{24} all primary contrasts.

Regime	Hypothesis	Gap	Cluster CI	Group p	p_H	p_{24}
TB	H1a-full	+44	[25.3, 56.2]	0.0054	0.0439	0.1235
TB	H1a-skill	+29	[12.8, 40.0]	0.0259	0.1812	0.4917
TB	H1a-hybrid	+32	[19.3, 44.0]	0.0049	0.0439	0.1172

Regime	Hypothesis	Gap	Cluster CI	Group p	p_H	p_{24}
TB	H1b-full	+25	[12.9, 37.6]	0.0078	0.0703	0.1719
TB	H1b-skill	+22	[8.0, 39.0]	0.0203	0.1418	0.4053
TB	H1b-hybrid	+18	[6.5, 32.5]	0.0137	0.1094	0.2871
TB	H2a	+11	[1.3, 18.7]	0.0859	0.4297	1.0000
TB	H2b	+0	[-6.2, 6.1]	1.0000	1.0000	1.0000
State	H1a-full	-1	[-7.7, 5.3]	1.0000	1.0000	1.0000
State	H1a-skill	+3	[-5.1, 10.7]	0.6367	1.0000	1.0000
State	H1a-hybrid	+4	[-5.8, 13.5]	0.5732	1.0000	1.0000
State	H1b-full	-3	[-13.3, 5.6]	0.6953	1.0000	1.0000
State	H1b-skill	-2	[-10.4, 6.9]	0.8281	1.0000	1.0000
State	H1b-hybrid	+1	[-7.2, 9.8]	1.0000	1.0000	1.0000
State	H2a	+4	[-5.2, 13.3]	0.6250	1.0000	1.0000
State	H2b	-3	[-13.1, 6.4]	0.7031	1.0000	1.0000
Text	H1a-full	+0	[-7.4, 6.2]	1.0000	1.0000	1.0000
Text	H1a-skill	+1	[-9.8, 13.2]	1.0000	1.0000	1.0000
Text	H1a-hybrid	-8	[-16.7, 2.0]	0.1915	1.0000	1.0000
Text	H1b-full	-2	[-14.3, 9.5]	0.8794	1.0000	1.0000
Text	H1b-skill	-2	[-10.1, 7.4]	0.8281	1.0000	1.0000
Text	H1b-hybrid	-7	[-15.6, 2.2]	0.2412	1.0000	1.0000
Text	H2a	+12	[3.1, 22.4]	0.0469	0.2812	0.8438
Text	H2b	-7	[-19.8, 4.2]	0.3613	1.0000	1.0000

B.3 Exploratory representation comparisons

The 36 within-branch tests cross six unordered representation pairs with U/C and three regimes. They ask which handoff is preferable within a fixed branch. For two formats $r, q \in \{\mathrm{None}, \mathrm{Full}, \mathrm{Skill}, \mathrm{Hybrid}\}$, the separate 18-test family compares $\widehat{\Delta}^{\mathrm{corr}}_{d,r} - \widehat{\Delta}^{\mathrm{corr}}_{d,q}$, and therefore answers a different question. No within-branch result survives group-level Holm over 36; three ThinkingBox C-versus-None comparisons survive the target-level version. Only ThinkingBox Full–None survives the group-level 18-test interaction family. These comparisons were motivated by inspecting the existing results and are exploratory.

Table 9: Within-branch comparisons. N/F/S/H: None/Full/Skill/Hybrid. G/L denotes observed upward/downward transitions. The last columns apply Holm across all 36 tests.

Regime	Comparison	G/L	Gap	Cluster CI	Target p_H	Group p_H
TB	U-F–U-N	4/18	-14	[-27.2, -4.3]	0.1433	0.7734
TB	U-S–U-N	12/14	-2	[-10.8, 4.8]	1.0000	1.0000
TB	U-H–U-N	9/18	-9	[-25.4, 3.3]	1.0000	1.0000
TB	U-S–U-F	16/4	+12	[2.0, 22.9]	0.3782	1.0000
TB	U-H–U-F	10/5	+5	[-4.3, 12.9]	1.0000	1.0000
TB	U-H–U-S	8/15	-7	[-20.7, 3.6]	1.0000	1.0000
TB	C-F–C-N	35/4	+31	[17.7, 42.4]	<.0001	0.1709
TB	C-S–C-N	34/6	+28	[12.4, 42.1]	0.0003	0.6143
TB	C-H–C-N	29/5	+24	[14.6, 32.5]	0.0013	0.1230
TB	C-S–C-F	10/13	-3	[-12.7, 7.5]	1.0000	1.0000
TB	C-H–C-F	5/12	-7	[-14.3, 3.0]	1.0000	1.0000
TB	C-H–C-S	13/17	-4	[-12.8, 6.8]	1.0000	1.0000
State	U-F–U-N	9/12	-3	[-12.7, 5.5]	1.0000	1.0000
State	U-S–U-N	9/15	-6	[-13.7, 2.1]	1.0000	1.0000
State	U-H–U-N	7/11	-4	[-10.5, 3.3]	1.0000	1.0000
State	U-S–U-F	12/15	-3	[-14.4, 8.5]	1.0000	1.0000
State	U-H–U-F	9/10	-1	[-9.8, 10.5]	1.0000	1.0000
State	U-H–U-S	11/9	+2	[-6.6, 10.8]	1.0000	1.0000
State	C-F–C-N	10/13	-3	[-14.1, 7.5]	1.0000	1.0000
State	C-S–C-N	9/11	-2	[-8.8, 4.7]	1.0000	1.0000
State	C-H–C-N	7/6	+1	[-6.0, 7.7]	1.0000	1.0000
State	C-S–C-F	12/11	+1	[-9.0, 11.7]	1.0000	1.0000
State	C-H–C-F	12/8	+4	[-8.2, 16.1]	1.0000	1.0000
State	C-H–C-S	9/6	+3	[-4.3, 9.7]	1.0000	1.0000

Regime	Comparison	G/L	Gap	Cluster CI	Target p_H	Group p_H
Text	U-F–U-N	10/10	+0	[-7.3, 7.4]	1.0000	1.0000
Text	U-S–U-N	11/12	-1	[-10.8, 7.1]	1.0000	1.0000
Text	U-H–U-N	11/8	+3	[-6.0, 11.7]	1.0000	1.0000
Text	U-S–U-F	11/12	-1	[-11.0, 8.0]	1.0000	1.0000
Text	U-H–U-F	13/10	+3	[-6.4, 12.5]	1.0000	1.0000
Text	U-H–U-S	9/5	+4	[-2.6, 10.8]	1.0000	1.0000
Text	C-F–C-N	9/11	-2	[-12.0, 7.1]	1.0000	1.0000
Text	C-S–C-N	10/12	-2	[-10.2, 6.6]	1.0000	1.0000
Text	C-H–C-N	5/12	-7	[-16.0, 2.0]	1.0000	1.0000
Text	C-S–C-F	12/12	+0	[-9.7, 10.8]	1.0000	1.0000
Text	C-H–C-F	4/9	-5	[-11.3, 2.2]	1.0000	1.0000
Text	C-H–C-S	7/12	-5	[-12.9, 2.4]	1.0000	1.0000

Table 10: Difference in correction gaps between representations. All intervals are pp; group p_H adjusts the 18 interactions.

Regime	Format contrast	Gap difference	Cluster CI	Group p_H
TB	Full–None	+45	[27.8, 61.9]	0.0352
TB	Skill–None	+30	[11.2, 48.6]	0.3945
TB	Hybrid–None	+33	[14.9, 53.9]	0.0623
TB	Skill–Full	-15	[-28.7, -1.4]	1.0000
TB	Hybrid–Full	-12	[-22.6, 2.4]	1.0000
TB	Hybrid–Skill	+3	[-11.3, 21.4]	1.0000
State	Full–None	+0	[-9.6, 8.8]	1.0000
State	Skill–None	+4	[-5.8, 12.7]	1.0000
State	Hybrid–None	+5	[-6.4, 15.2]	1.0000
State	Skill–Full	+4	[-4.2, 10.6]	1.0000
State	Hybrid–Full	+5	[-8.0, 17.3]	1.0000
State	Hybrid–Skill	+1	[-12.1, 12.9]	1.0000
Text	Full–None	-2	[-10.1, 4.9]	1.0000
Text	Skill–None	-1	[-12.3, 12.5]	1.0000
Text	Hybrid–None	-10	[-20.0, 1.0]	1.0000
Text	Skill–Full	+1	[-13.0, 16.7]	1.0000
Text	Hybrid–Full	-8	[-18.9, 4.3]	1.0000
Text	Hybrid–Skill	-9	[-20.6, 2.2]	1.0000

C RQ1: Benefit, Harm, Heterogeneity, and Target Errors

C.1 Complete B-only/U/C outcome decomposition

With bit order B-only/U/C, 001 is a new success beyond both references, 101 restores a baseline success lost under U, 010 loses U's help, and 110 breaks a shared B-only/U success. Their signed sum is the C–U effect. The other four patterns do not change between U and C. These are observed outcome paths rather than individual causal effects.

Table 11: All eight outcome patterns, including unchanged outcomes. Each column sums to 100. F/S/H denote Full/Skill/Hybrid.

	ThinkingBox			APEX-state			APEX-text		
B/U/C	F	S	H	F	S	H	F	S	H
000	26	22	31	43	40	42	38	37	38
001	24	21	16	4	7	6	5	5	2
010	1	6	3	3	4	3	4	7	8
011	7	9	8	6	5	5	6	4	5
100	5	4	5	7	6	5	7	5	9
101	22	18	20	4	8	6	5	9	3
110	1	4	1	6	8	5	6	6	5
111	14	16	16	27	22	28	29	27	30

Conditional improvements and partial credit. Let $G = n_{001} + n_{101}$ count gains, $L = n_{010} + n_{110}$ losses, and $Y_i^U = Y_{di}^{U,r}$. The upward proportion among U failures is $G/(N - \sum_i Y_i^U)$; the downward proportion among U successes is $L/\sum_i Y_i^U$. Full improves 46/77 U failures in ThinkingBox (59.7%), 8/58 in APEX-state (13.8%), and 10/55 in APEX-text (18.2%). It loses 2/23 U successes (8.7%), 9/42 (21.4%), and 10/45 (22.2%), respectively. APEX therefore has substantial headroom, but recovers a smaller fraction of failures while losing a larger fraction of successes.

Aligned rubric differences for Full/Skill/Hybrid are $-4.90/+1.67/+1.29$ pp in APEX-state and $-2.32/+0.36/-8.26$ pp in APEX-text. Text Hybrid gains 5.35 pp of task-weighted item credit but loses 13.61 pp. Text Skill improves 49 raw rubric items and loses 34, yet gains only 0.36 pp after weighting tasks equally. Counting improved items alone would overstate target-level utility.

C.2 Relationship and domain heterogeneity

Request-based relationship coding. The 100 frozen pairings shared by both APEX regimes are categorized by reading the complete A and B requests. The mutually exclusive categories use the precedence in Table 12: the same substantive operation takes priority over a shared intermediate procedure, which takes priority over common evidence. Merely using spreadsheets, legal reasoning, or the same application is insufficient to establish a shared procedure. Counts are numbers of pairs, not numbers of distinct families.

Table 12: Relationship definitions and examples from the request-based audit.

Relationship	Pairs	Shared basis	What changes in B
Same-family variant	42	Same computation, model, or legal decision test	LBO debt/rate assumptions; a different insured claim
Shared intermediate procedure	26	Identifiable reusable subprocedure; distinct end operation	Comparable filtering for peer benchmarking versus LBO inputs
Shared evidence	20	Common case materials or model; distinct operations	Manual drafting versus compliance comparison
Context only	12	Same project or domain; no specific shared procedure or material established	Board-tenure comparison versus CAPM estimation

This is a retrospective, single-pass annotation, not a preregistered or independently blinded classification. Aggregate results were already known; no per-cell target outcomes, gold answers, or rubric scores were used to assign labels. Each pair's shared basis and target change are recorded in `data/apex_pair_audit.json`. Borderline abstractions remain judgment-dependent, and no independent agreement estimate is available. The 42 same-family pairs comprise 26 banking, ten legal, and six consulting pairs; relationship and domain are therefore not independently varied.

Table 13: Matched APEX relationship strata. Every effect is shown as **state / text** in percentage points. Full/Skill/Hybrid are C–U; E–S is Execution–Summary.

Relationship	Count	Full	Skill	Hybrid	E–S
Same-family	42	-4.8 / +0.0	+7.1 / -7.1	-2.4 / -9.5	+2.4 / +14.3
Shared procedure	26	+0.0 / +0.0	+0.0 / +19.2	+19.2 / -3.8	+3.8 / +3.8
Shared evidence	20	-5.0 / +5.0	-5.0 / -5.0	-5.0 / -5.0	+15.0 / +15.0
Context only	12	+16.7 / -8.3	+8.3 / +0.0	+8.3 / -16.7	-8.3 / +16.7

Table 14: Domain-specific contrasts, again **state / text** in percentage points. Same definitions as Table 13; these are exploratory subgroup results.

Domain	Count	Full	Skill	Hybrid	E–S
Banking	48	-10.4 / -6.2	+0.0 / -6.2	+0.0 / -18.8	+14.6 / +10.4
Law	23	+4.3 / +0.0	+17.4 / +30.4	+17.4 / +4.3	-8.7 / +21.7
Consulting	29	+10.3 / +10.3	-3.4 / -10.3	+0.0 / +0.0	-3.4 / +6.9

Let N_k count the pairs in stratum k of a given partition (relationship or domain). The whole-cohort difference is the weighted sum of stratum differences, with weight N_k/N and $N = 100$. APEX-

state Hybrid's four net successes comprise -1 in same-family pairs, $+5$ in shared procedures, -1 in shared evidence, and $+1$ in context-only pairs. This locates offsetting changes rather than identifying a causal effect of relationship type. Shared procedures span 12 worlds (11 banking, two law, and 13 consulting pairs); shared evidence spans 11 worlds (six banking, eight law, and six consulting pairs).

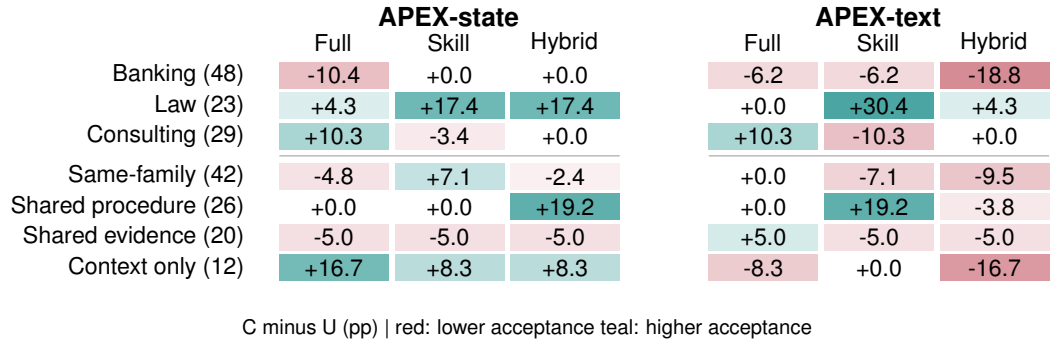


Figure 6: APEX C–U differences (pp); parentheses: subgroup sizes. Domains and relationships separately partition the same 100 pairs. Color describes differences, not significance.

Among the 42 same-family pairs, B-only solves twenty in each regime. State C-Full/Skill/Hybrid solve 14/17/16, versus 21/19/18 for text. Text Full's seven additional successes are not a correction gain: its U and C both solve 21. In shared procedures, state Hybrid and text Skill each gain five successes, whereas text Hybrid loses one. Relatedness provides an opportunity for reuse, not a guarantee that a source choice applies.

Legal Skill moves from eight to twelve successes with state and from four to eleven with text. The larger text correction gap reflects a weaker U, not a stronger C. Banking Hybrid moves from twenty to twenty with state and from 26 to seventeen with text, with no gains and nine losses. This combines six additional U successes with three fewer C successes; it is not nine C failures attributable to removing state. The text banking contrast has raw $p = 0.003906$ and twelve-test domain-family $p_H = 0.046875$; subgroup analyses remain exploratory.

C.3 Consistency across representations and handoff regimes

Table 15: Targets improved or regressed under at least one, at least two, or all three C–U representations. Improvement and regression columns are not disjoint: a target can improve under one representation and regress under another.

	Improved			Regressed		
Regime	≥ 1	≥ 2	All 3	≥ 1	≥ 2	All 3
ThinkingBox	61	42	18	15	1	0
APEX-state	29	5	1	22	6	1
APEX-text	22	6	1	28	7	1

Table 16: Matching the *same target's contrast* across APEX regimes. Gains/losses mean C–U, except E–S. "G to L" means a state gain becomes a text loss; "L to G" is the reverse. Each row has 100 matched pairs.

Contrast	State gains	Text gains	Gain in both	Loss in both	G to L	L to G
Full	8	10	2	2	0	0
Skill	15	14	5	5	3	2
Hybrid	12	5	2	4	4	0
Execution–Summary	12	14	2	0	0	4

APEX B-only passes in both regimes for 37 targets, in state only for seven, and in text only for ten; 46 fail in both. Even without any A-derived handoff, observed outcomes change across executions. The matched contrasts compare correction-associated changes over and above each regime's own baseline, rather than treating a difference between two condition means as a state effect.

Table 17: Within-regime effects and matched differences of effects. The last column is a world-cluster 95% interval for text minus state, in percentage points.

Contrast	State	Text	Difference	World interval
C-Full–U-Full	-1	0	+1	$[-6.19, 6.67]$
C-Skill–U-Skill	+3	+1	-2	$[-15.74, 12.77]$
C-Hybrid–U-Hybrid	+4	-8	-12	$[-24.00, 1.10]$
Execution–Summary	+4	+12	+8	$[-5.46, 24.27]$
Full–Execution	-3	-7	-4	$[-19.80, 7.56]$

The matched gap difference $\widehat{\delta}_r = \widehat{\Delta}^{\text{corr}}_{\text{text},r} - \widehat{\Delta}^{\text{corr}}_{\text{state},r}$ compares correction effects on the same 100 pairs. It is $+1/-2/-12$ pp for Full/Skill/Hybrid, with every world interval spanning zero (Table 17). It compares gaps, not C endpoints: separate significance does not establish an interaction. Text construction and execution batches also differ, so this comparison does not isolate state inheritance.

Similar totals need not identify the same beneficiaries. Only two of state Full's eight gains overlap with text Full's ten; Skill shares five gains and Hybrid two. Hybrid has four targets changing from a state gain to a text loss, with none in the reverse direction. Full has no direct gain-to-loss reversal: its differences involve gains or losses becoming ties, or vice versa (Table 16).

C.4 Cohort-wide validator profiles and cluster influence

We export the saved validator paths for all 1,100 ThinkingBox cells, including successes, from the same files used for acceptance. A lifecycle flag marks a path beginning with `zendesk_ticket_status_violations`; a type flag marks `zendesk_tickets[k].type`. All failed cells have recorded difference paths. These two diagnostics are overlapping subsets of benchmark errors, not an exhaustive failure taxonomy. The flags refer to the target validator; they do not label whether the source feedback explicitly taught that requirement.

Table 18: Recorded lifecycle violations, all 100 paired targets in each comparison. "Removed" counts reference-flagged targets without the flag in the candidate. Its last two columns partition those targets by candidate acceptance.

Comparison	Before	After	Removed	Then pass	Still fail
U-Full to C-Full	55	2	54	37	17
U-Skill to C-Skill	40	7	39	28	11
U-Hybrid to C-Hybrid	48	3	48	30	18
Summary to Execution	13	1	13	8	5

New flags also appear: one for C-Full, six for C-Skill, three for C-Hybrid, and one for Execution, relative to their respective references. Thus removing flags on some targets is compatible with introducing them elsewhere. The identity "removed = then pass + still fail" distinguishes correcting a recorded local error from satisfying every task requirement. It does not estimate a causal mediation effect. `data/validator_profiles.json` retains every pair's paths; `data/validator_analysis.json` includes both flag types and the paired assignments.

ThinkingBox's C–U advantage remains positive after omitting any one family. Both APEX regimes retain substantially smaller effects. Text Execution–Summary remains positive after omitting any one world; this influence result is distinct from multiplicity-adjusted testing.

Table 19: Leave-one-cluster-out acceptance gaps (pp), including all three regimes. Each row removes one complete family or world at a time. Min/max are influence ranges, not confidence intervals. Abbreviations follow Table 7.

Regime	Comparison	Clusters	Minimum	Maximum
ThinkingBox	C-F / U-F	15	+40.66	+48.42
ThinkingBox	C-S / U-S	15	+25.27	+31.96
ThinkingBox	C-H / U-H	15	+27.78	+34.74
ThinkingBox	E / S	15	+8.64	+13.33
APEX-state	C-F / U-F	16	-2.22	+1.08
APEX-state	C-S / U-S	16	+1.05	+4.30
APEX-state	C-H / U-H	16	+1.06	+6.19
APEX-state	E / S	16	+0.00	+5.43
APEX-text	C-F / U-F	16	-1.14	+2.13
APEX-text	C-S / U-S	16	-1.08	+4.55
APEX-text	C-H / U-H	16	-10.53	-5.68
APEX-text	E / S	16	+8.51	+13.98

D Action-Level Evidence for Transfer and Its Limits

ThinkingBox: a reusable constraint can outlive the source decision. U-Full records 55 ticket-lifecycle violations; C-Full records two. Of 54 reference-flagged targets without that flag in C, 37 pass and 17 still fail; one new flag appears elsewhere. This distinguishes improvement in an operational requirement from full target acceptance. In `lif_017`→`lif_015`, the business eligibility decision changes, but the open–act–close requirement remains. B-only and U-Skill correctly decline B while creating already-solved tickets; C-Skill declines it with the required lifecycle. The source concerns an unpaid Preferred-tier customer eight days after cancellation, within a 30-day reinstatement window; B concerns a paid customer 35 days after cancellation. All inspected B-only/U-Skill/C-Skill runs correctly decline B, separating the changed business decision from the reusable ticket workflow. By contrast, a source correction from Daughter to Child in `drv_101`→`drv_008` cannot be claimed as field-level transfer when B rejects the request and never writes that field. Episode usefulness and transfer of a specific correction are different claims.

APEX: a source-convention conflict occurs under both handoffs. For `ad52b8019140`, B-only and all three U representations pass in *both* regimes, while all three C representations fail with zero rubric credit. B explicitly specifies mid-year convention. In the text trace, C-Skill labels full-year terminal-value discounting as user-validated. B reads the original model, calculates both six-year and 5.5-year alternatives, then selects the source-validated six-year result: 40.45/44.83 instead of B's fixed-rubric 41.62/46.22. The observed conflict concerns applicability and selection, not missing retrieval or calculation. The same score pattern occurs without inherited A files; the detailed selection chain is established from the text run. It does not isolate the approval marker's causal contribution.

Positive transfer can mix a method with a source conclusion. For `327f2507ef39`, B-only and all U variants fail in both regimes. State C-Full and C-Skill pass while C-Hybrid fails; all three text C variants pass. C-Skill prescribes a requirements–source-clause matrix, entity-role separation, and avoiding defects imported from unrelated documents. It also retains A's conclusion that the reviewed documents comply. B uses the matrix and reaches the rubric's compliant conclusion. This is useful corrected experience in both regimes, with representation-sensitive coverage; it does not separate procedural learning from answer anchoring.

Restoration and loss of U's help are also visible in both regimes. Helios (`993a38dad720`) has B-only success and a U-Hybrid failure followed by C-Hybrid success in both state and text. The text artifact audit identifies the changed data scope: U unions 400-, 2,000-, and 2,500-row inputs into 4,900 SKUs and reports a 22.06% VW share; C uses the specified 2,500-row Rebuilt input, with 541 VW entries and 21.64%. This restores an independently achievable result by applying an appropriate scope constraint.

Lumea (`e8b80949477a`) shows the opposite result in both regimes: U-Skill and U-Hybrid pass, both corresponding C conditions receive 1/3, and B-only also receives 1/3. The text trace computes

alternatives from workbook precision and rounded CSV values but selects the latter. Weighted margins are 56.786890% versus 56.795421%; the 30% improvement opportunity is 607,500.34 versus 606,666.14. A shared displayed margin of 56.8% hides a consequential precision choice. This is a reference-sensitive loss relative to U, not a loss of observed baseline success; the score alone does not establish harmful reasoning.

These cases expose a common decision point: **which source constraint should govern the target?** Correct information can be available but unenacted; an enacted rule can be inappropriate; a corrected local behavior can still leave other requirements unsatisfied. The observed ThinkingBox benefit is compatible with reusable workflow constraints, whereas APEX cases expose version, scope, precision, and convention selection. We do not infer a benchmark-wide causal explanation from those cases.

State availability versus correction benefit. Physical state changes what must be reconstructed, but does not decide which result is applicable. In Planet Fitness (`01da0d93b3ad`), the source text names v08 while the text target has v06; Summary and Execution check availability and must rebuild missing work. In the core conditions, state U/C-Full both fail, while text U/C-Full both pass. This level change creates no C–U improvement in either regime. Thus successful reconstruction is not automatically evidence of correction transfer.

ThinkingBox disagreement set. All 15 Execution gains against Summary and all four losses are included in `data/case_evidence.json`, with task IDs and validator paths from the failed execution. The eight gains with ticket-status violations are `veh_013`, `bil_016`, `drv_021`, `drv_101`, `veh_005`, `bil_014`, `exp_011`, and `hwa_013`. Ticket-type mismatches occur in `ldr_008` and `bil_010`. The other gains are `bil_008`, `trv_009`, `csa_004`, `exp_002`, and `onb_007`. Secondary timestamp differences are not counted as separate explanatory mechanisms. The four losses are `lif_012` (incorrect billing-state change), `swa_013` and `onb_011` (missing approval fields), and `doc_001` (ticket-status violation). The coding is based on outcome-selected validator differences, not blinded causal annotation.

Insurance rejection: business decision versus operational contract. The `ldr_008` target belongs to `sandbox_auto_insurance_group1.py`. In both executions the agent rejects a billing change for a listed driver lacking policyholder authority. The supplied source summary explicitly states "type task". Nevertheless, the Summary-condition target updates the ticket to a solved rejection without setting its type; Execution sets type to task and completes its status update. The other ticket-type gain, `bil_010`, also receives a summary explicitly specifying "type task". Both errors occur despite that field value being available, rather than because compression removed it. `data/case_evidence.json` includes the supplied summary and source task for all 19 disagreements. These comparisons establish an availability–enactment distinction; they do not isolate whether repetition, action formatting, or another feature of the added trace drives it.

APEX-state valuation: a common upstream source affects four criteria. Pair `apex_full_3724da87dc96` requests a downside deck for 3M's 20% stake using a 10% free-cash-flow reduction. The following values are taken from delivered answers and rubric evidence. Summary uses the `v1 - FIXED` workbook, whereas Execution's primary source is the original `v1.xlsx`, `DCF-Solv` tab. Execution also computes the corrected copy as a labeled cross-check. C-Full yields 1/5 and a current stake of approximately $4,936.4M using the corrected

Table 20: 3M downside analysis. Dollar values are millions. All conditions produce a deck; model lineage changes the numerical findings.

Requested quantity	Summary	Execution	Reference
Current stake	4,935.9	5,499.7	5,499.7
PV of revised cash flows	4,389.4	4,790.1	4,790.1
Discounted terminal value	17,822.9	19,959.1	19,959.0
Revised stake	4,442.5	4,949.8	4,949.8
Percentage loss	10.0%	10.0%	10.0%
Criteria satisfied	1/5	5/5	–

lineage. B requests values "directly from the accretion dilution model," a 20% ownership stake, and a 10% reduction in 2025–2029 cash flows, but does not name the original or corrected version. The saved reads identify a lineage-dependent score difference; they do not establish which interpretation a user would intend after A's accepted repair. The fixed-reference outcome is retained without treating it as proof that correction history harms reasoning.

Distinguishing inherited errors from new target errors. In a separate ThinkingBox equipment case, `hwa_013`, the C-Skill run assigns its second reserved device rather than its first. Neither identifier appears in the source handoff: both originate in actual B-stage tool returns. The first reservation remains orphaned and a ticket-status violation is recorded. This is a new execution-side divergence, not evidence that memory copied a stale identifier. The packaged supplementary evidence records this provenance.

D.1 Cross-family evidence and ambiguous score differences

Shared-procedure APEX-state Hybrid has six C–U gains and one loss (raw $p = 0.125$), compared with four gains and one loss against B-only ($p = 0.375$). In peer-screening pair `4e38ae1e886d`, both branches exclude the same comparables and calculate the same 20.67x median, 30.67x exit multiple, and 3.42x debt multiple. The difference arises later in LBO returns, so it does not demonstrate learning the common screening step. In `432cf30bbe45`, review explanations invoke different acquisition-rate targets (4.9% versus 4.1% after tax); omitting that pair leaves five gains and one loss among 25 pairs (+16.0 pp, raw $p = 0.219$).

All three shared-evidence Execution–Summary gains were inspected. In Kenvue (`6fa5694c8bbe`), Execution selects volume-weighted closing prices while Summary selects arithmetic means, changing rubric satisfaction from 3/5 to 5/5. Summary also computes weighted alternatives, and neither source handoff teaches VWAP. In Aptar (`f58e573d229d`), both answers headline 36.40% and provide 28.61 points under an alternative interpretation. In purchase-price allocation (`32223647d439`), Execution's review marks goodwill passed while its evidence describes a mismatch. The latter two are not behaviorally corroborated gains; omitting both leaves one gain and no loss among 18 pairs (+5.6 pp, raw $p = 1$). Original outcomes remain intact.

These selected audits distinguish transferred procedure, source-answer anchoring, target-side evidence selection, and judge disagreement. They are evidence about particular action chains, not estimates of the prevalence of each mechanism.

E RQ2 and RQ3: Accepted Execution and Resource Tradeoffs

Table 21: Accepted-experience conditions. G/L counts paired gains/losses for Execution–Summary; p_H is the six-test H2-family adjustment.

Regime	Summary	Execution	Full	G/L	E–S	p_H
ThinkingBox	56	67	67	15/4	+11	0.0961
APEX-state	40	44	41	12/8	+4	1.0000
APEX-text	40	52	45	14/2	+12	0.0251

E.1 Resource thresholds and initial payloads

Execution reduces mean cumulative input relative to Full by 18.7% in ThinkingBox, 12.1% in APEX-state, and 31.8% in APEX-text. Mean model calls instead increase by 4.7%, 105.7%, and 16.9%. H3a is consistent across observed means; H3b has the opposite direction throughout. The text cost analysis gives a world interval of $[-1.607, -0.728]$ million input tokens and $[1.10, 7.94]$ additional calls, with two-test H3 Holm values 0.0002 and 0.0068.

The accounting is multiplicative: cumulative input equals calls times average input per call. In state APEX, Execution roughly halves input per call while doubling calls, leaving only a 12.1% net input reduction. Text APEX reduces input per call from 119.2k to 69.5k, enough to offset its smaller

Table 22: B-stage resources for B-only and corrected handoffs. Each mean includes all 100 targets, including failures. Input is cumulative millions of tokens; input/call is total input divided by total calls in thousands.

Regime	Condition	Accepted	Input M	Calls	Input/call k	Tools
ThinkingBox	B-only	42	0.232	9.48	24.4	14.96
ThinkingBox	Summary	56	0.264	9.48	27.9	17.09
ThinkingBox	Execution	67	0.353	9.27	38.1	18.34
ThinkingBox	C-Full	67	0.434	8.85	49.0	17.07
ThinkingBox	C-Skill	64	0.286	9.80	29.2	17.61
ThinkingBox	C-Hybrid	60	0.318	9.82	32.4	17.44
APEX-state	B-only	44	2.189	37.78	58.0	46.47
APEX-state	Summary	40	2.307	43.01	53.6	56.51
APEX-state	Execution	44	2.405	41.22	58.4	54.92
APEX-state	C-Full	41	2.737	20.04	136.6	22.83
APEX-state	C-Skill	42	2.337	39.19	59.6	48.50
APEX-state	C-Hybrid	45	2.273	37.39	60.8	46.46
APEX-text	B-only	47	2.227	37.51	59.4	46.19
APEX-text	Summary	40	2.172	37.29	58.3	45.72
APEX-text	Execution	52	2.438	35.08	69.5	44.40
APEX-text	C-Full	45	3.576	30.01	119.2	38.24
APEX-text	C-Skill	45	2.383	40.46	58.9	48.97
APEX-text	C-Hybrid	40	2.236	37.22	60.1	45.75

call increase. Tool calls in text rise from 38.24 to 44.40, while output tokens remain approximately 56k. A shorter handoff changes how much context is repeatedly read and how much work remains downstream; it does not uniformly shorten execution.

The cost interpretation also depends on the reference. Text Execution uses 12.2% more input than Summary (2.438M versus 2.172M), while its 2.21-call reduction has a world interval crossing zero. Skill and Hybrid reduce input relative to Full but require more calls and do not dominate its acceptance. At an observed completed-trajectory ceiling of two million input tokens, text Execution has 32 accepted targets versus Full's 11; at twenty model calls, Full has 18 versus Execution's ten. These are counts of completed accepted runs within resource bounds, not early-stopping experiments.

Input includes cached context, so token reductions are not billing reductions. Initial payload and whole initial-message length are also distinct: the text source handoff averages 281.2k characters for Full versus 45.3k for Execution, while serialized initial messages average 289.7k and 48.8k. Costs cover the current B attempts, excluding source correction, memory construction, judging, and historical infrastructure retries.

F Judging, Coverage, and Sensitivity Analyses

F.1 Full-denominator coverage and artifact reconstruction

Table 23: APEX execution coverage. Completed means agent submission, not owner acceptance. Budget and infrastructure failures remain distinct. Every cell has a judgment in the aligned analysis.

Regime / set	Completed	Budget failed	Error	Total
State: original nine conditions	895	5	0	900
State: Summary	92	8	0	100
State: Execution	88	10	2	100
Text: all eleven conditions	1,072	28	0	1,100

State judgments reconstruct the original world, A inputs and branch delta, B inputs, and B delta. Text judgments reconstruct the original world, B inputs, and the text run's B delta without restoring A. Both apply the original target rubric to native visual evidence and available artifacts. Technical rendering, transport, or response-format errors are not scored as task failures. Valid judgments are retained rather than repeated because of a low score.

F.2 Partial-artifact alignment and execution-error sensitivity

The original state archive contains 1,057 primary `deepseek-flash` reviews, 23 `gpt-5.6-luna` fallback reviews, and 20 policy-zero cells. The latter comprise 18 budget failures and two execution errors. An independent alignment layer reviews their saved partial artifacts with `deepseek-flash`; all 20 remain unaccepted. Summary's mean rubric score changes from 49.22% to 49.65%; Execution remains 52.33%. The current condition table uses these aligned rubric values. Original reviews, policy-zero scores, and execution identities are preserved.

For H2 contrasts involving Execution, excluding the two state execution-error pairs leaves 98 matched pairs. Execution–Summary is $+5.10$ pp in state and $+12.24$ pp in text, a matched regime difference of $+7.14$ pp, compared with $+8$ pp over all 100. Full–Execution is -4.08 pp and -7.14 pp, respectively, a difference of -3.06 pp. These exclusions do not convert the regime comparison into a randomized state intervention.

F.3 Backend and disagreement-audit sensitivity

APEX-text has 1,083 valid primary reviews and 17 `gpt-6-luna` fallbacks for technical failures. Sixteen fallback cells retain verified native-image tool traces; the first valid fallback has incomplete image-access audit provenance. Among the 98 pairs with primary-backend reviews for both Summary and Execution, the gap is $+12.24$ pp (14 gains, two losses). This subset is not evidence that judge backends are interchangeable.

The complete 16-case APEX-text Execution–Summary disagreement audit separates requirement coverage, formal delivery, answer priority, numerical precision, and scoring contradictions. Four clearly inconsistent judgments are retained as a separate sensitivity layer. Applying those audit corrections gives 50/38 rather than 52/40 accepted targets, leaving a twelve-point gap but changing 14/2 discordances to 12/0. Main results retain the first valid judgments. Because this audit selects disagreements and is not condition-blinded, it does not estimate overall judge accuracy or provide a second independent test.

Text Execution–Summary also improves rubric satisfaction by 8.96 pp; its adjusted $p = 0.0568$ is from the separate eight-contrast text rubric family, not the acceptance families in Table 7. Acceptance, partial credit, backend restriction, and action-level audit answer different questions and are reported separately.

G Extended Literature Comparison

Correction in continuing human–AI collaboration. Amershi et al. (2019) distinguish making errors easy to correct, remembering recent interaction, and adapting from user behavior over time. Our question concerns their intersection: does an accepted repair improve a new case, or must its lesson be supplied again? We operationalize this through paired agent executions with simulated feedback, rather than a human-subject evaluation of interface usability or oversight effort.

Improving an attempt versus improving the next task. Self-Refine (Madaan et al., 2023) improves outputs through self-feedback; Reflexion (Shinn et al., 2023) turns task feedback into reflections for later attempts. Self-correction without external feedback need not improve reasoning (Huang et al., 2024), and early mistaken assumptions can persist across turns (Laban et al., 2025). We examine what happens *after* A is accepted: how corrected and uncorrected experience affect a distinct B.

Evidence for cross-task experience reuse. ExpeL (Zhao et al., 2024) extracts knowledge from training-task experience for inference-time retrieval. CLIN (Majumder et al., 2024) updates causal abstractions and cross-episode meta-memory for new tasks and environments. Agent Workflow Memory (AWM) (Wang et al., 2025) induces workflows for offline, online, and cross-domain use on Mind2Web and WebArena. ReasoningBank (Ouyang et al., 2026) distills self-judged successes and failures; memory-aware test-time scaling (MaTTS) expands exploration and ablates failed-trajectory contributions. We do not propose a larger skill library or a new memory learner. We pair one failed

episode with its accepted repair and hold B fixed, asking what changes downstream when the source is corrected. Summary and execution controls then locate how the corrected experience is conveyed.

Skills can also hurt. Trace2Skill (Ni et al., 2026) aggregates trajectory-local lessons and tests cross-model and out-of-distribution transfer. *Break It Down, Pass It On* (Feng et al., 2026) compares induction granularity and text/code formats: task-level skills often hurt, while subtask-level skills help on average, with specificity and abstractness characterizing utility. Its analyses also examine source outcomes and intervene on skill-library utility; applicability and negative transfer are not unique to our study. SkillsBench (Li et al., 2026a) pairs no-skill and curated-skill conditions, also tests self-generated skills, and analyzes gains and regressions. Our added axis pairs the *same episode before and after accepted repair*, separating an update's incremental value from the corrected memory's value relative to independence (Table 24).

Compression and inference-time adaptation. LLMLingua (Jiang et al., 2023) compresses prompts; Lost in the Middle (Liu et al., 2024) studies positional effects in long-context use. The Complexity Trap (Lindenbauer et al., 2025) compares observation masking with model summaries, while Evaluating AGENTS.md (Gloaguen et al., 2026) measures benefits and costs of repository context files. Trace as State (Zou & Tang, 2026) prepends reasoning traces to support rereading within a task. We fix model parameters and study experience utility and execution cost across a task boundary.

Evaluation granularity and reliability. MemoryArena (He et al., 2026) evaluates memory-guided action across interdependent sessions; we additionally vary source correction. Micro-benchmarking examines ranking reliability under small gaps (Yauney et al., 2026). We decompose outcomes through acceptance, rubric satisfaction, and paired trajectories.

Table 24: Comparison units and experimental axes in cross-task experience reuse.

Study	How experience transfers	Main comparison axis
ExpeL (Zhao et al., 2024)	Knowledge extraction and experience retrieval	Accumulation and transfer
CLIN (Majumder et al., 2024)	Causal abstractions and meta-memory	Adaptation and generalization
AWM (Wang et al., 2025)	Induced reusable workflows	Offline/online and cross-domain
ReasoningBank (Ouyang et al., 2026)	Strategies from success and failure	Memory; exploration via MaTTS
Feng et al. (2026)	Task- or subtask-induced skills	Granularity $\times$ text/code
SkillsBench (Li et al., 2026a)	Curated or generated skills	Skill source, domain, and utility
This study	The same source failure with and without an accepted repair, transferred to a fixed B	Memory-update value versus predecessor and independence; repair $\times$ representation