PT Jurnal Cendekia Indonesia
Journal of Information Technology and Informatics Engineering
Homepage: https://journal.jci.co.id/jitie
Vol. 02 No. 04 (2026) Page: 59-75
E-ISSN: 3123-7827

Event-Sourced Memory and State-Consistency Verification for Dual-Control Enterprise Agents: Recovery, Rollback, and Human Escalation

James Jackson[1], Vincent Tan[2], Isabella White[3], Fiona Shi[4]

[1]Department of Computer Science, University of Arizona, Tucson, AZ, USA
[2]Department of Computer Science, North Carolina State University, Raleigh, NC, USA
[3]Department of Computer Science, Iowa State University, Ames, IA, USA
[4]Department of Computer Science, University of Delaware, Newark, DE, USA

Email: [1*]j.jackson_az@proton.me

Abstract

Enterprise language-model agents operate through conversations, mutable records, external tools, and user actions that can change the same environment. Reliability therefore depends on authoritative execution provenance, duplicate and conflict detection, boundary-resume behavior, rollback, and a state-verified escalation decision when automation cannot continue. This study evaluates four coupled controller-policy packages—Stateless, Full History, Summary Memory, and Event-Sourced Memory—on all 114 telecom tasks in the fixed τ²-bench v0.2.0 release. The telecom corpus contains 516 listed actions, including 393 user-side and 123 assistant-side actions, plus 209 environment assertions. An oracle-plan protocol holds semantic planning constant while native tools execute paired clean, recoverable-fault, and persistent-failure runs. The main campaign contains 13,960 runs and the component campaign contains 12,080 runs. Separately, 12 release-supplied primary result files provide 4,448 model trajectories across airline, retail, and telecom, with 56.83% pooled success. Across 2,450 recoverable-fault runs per controller, task success was 45.92% for Stateless, 89.55% for Full History, 80.37% for Summary Memory, and 100.00% for Event-Sourced Memory. The event package also recorded complete safe resolution, no unresolved failures, and no controller-induced violations. Under persistent failure, each stateful package made a state-verified escalation decision in every case; no person was invoked or scored. Removing the durable idempotency index, precondition policy, or rollback reduced recoverable success to 94.04%, 90.94%, and 82.33%. The results support separating semantic context from transaction-aware execution state while limiting claims to the evaluated policy packages and deterministic interventions..

Keywords: Large Language Models; Enterprise Agents; Event Sourcing; State Consistency; Dual Control; Tool Reliability; Idempotency; Rollback; Escalation; T²-Bench.

Abstrak

Agen model bahasa tingkat perusahaan beroperasi melalui percakapan, catatan yang dapat diubah, alat eksternal, dan tindakan pengguna yang dapat mengubah lingkungan yang sama. Oleh karena itu, keandalan bergantung pada asal-usul eksekusi yang otoritatif, deteksi duplikasi dan konflik, perilaku melanjutkan tugas dari titik henti (boundary-resume), pemulihan keadaan sebelumnya (rollback), serta keputusan eskalasi yang diverifikasi berdasarkan status saat otomatisasi tidak dapat dilanjutkan. Studi ini mengevaluasi empat paket pengendali-kebijakan (controller-policy)—Stateless, Full History, Summary Memory, dan Event-Sourced Memory—terhadap seluruh 114 tugas telekomunikasi dalam rilis τ²-bench v0.2.0 yang baku. Korpus telekomunikasi tersebut memuat 516 tindakan terdaftar, yang mencakup 393 tindakan sisi pengguna dan 123 tindakan sisi asisten, serta 209 pernyataan kondisi lingkungan. Protokol rencana oracle menjaga perencanaan semantik tetap konstan sementara alat bawaan menjalankan eksekusi berpasangan yang terdiri dari skenario bersih (clean), kesalahan yang dapat dipulihkan (recoverable-fault), dan kegagalan persisten (persistent-failure). Rangkaian pengujian utama mencakup 13.960 kali eksekusi, sedangkan rangkaian pengujian komponen mencakup 12.080 kali eksekusi. Secara terpisah, 12 berkas hasil utama yang disertakan dalam rilis menyediakan 4.448 lintasan model di berbagai sektor—penerbangan, ritel, dan telekomunikasi—dengan tingkat keberhasilan gabungan sebesar 56,83%. Pada 2.450 eksekusi dengan kesalahan yang dapat dipulihkan untuk setiap pengendali, tingkat keberhasilan tugas adalah 45,92% untuk Stateless, 89,55% untuk Full History, 80,37% untuk Summary Memory, dan 100,00% untuk Event-Sourced Memory. Paket berbasis event juga mencatat penyelesaian yang aman dan tuntas, tanpa kegagalan yang tidak teratasi, serta tanpa pelanggaran yang disebabkan oleh pengendali. Dalam kondisi kegagalan persisten, setiap paket yang menyimpan status (stateful) mengambil keputusan eskalasi yang diverifikasi berdasarkan status pada setiap kasus; tidak ada keterlibatan atau penilaian terhadap manusia. Penghapusan indeks idempotensi permanen, kebijakan prasyarat, atau fitur rollback menurunkan tingkat keberhasilan pemulihan masing-masing menjadi 94,04%, 90,94%, dan 82.33%. Hasil penelitian ini mendukung pemisahan konteks semantik dari

Received: 01-08-2026 | Revised: 09-08-2026 | Published Online: 10-08-2026

status eksekusi yang sadar transaksi, sembari membatasi klaim hanya pada paket kebijakan yang dievaluasi dan intervensi yang bersifat deterministik.

***Kata Kunci**: Model Bahasa Besar; Agen Perusahaan; Event Sourcing; Konsistensi Status; Kontrol Ganda; Keandalan Alat; Idempotensi; Rollback; Eskalasi; T²-Bench*

A. INTRODUCTION

Large language models increasingly mediate operational workflows such as account maintenance, device configuration, order management, and customer support. Unlike a question-answering system, an enterprise agent can modify durable records through tools while a user, background process, or another agent changes the same environment. A plausible response is therefore insufficient evidence of correctness. The workflow must execute in a permitted order, distinguish retries from new intent, avoid repeating committed effects, and stop at a known state when automatic recovery is exhausted.

τ²-bench formalizes this problem through dual-control tasks in which the conversational agent and simulated user possess different tools and jointly affect a shared environment [1]. Its predecessor, τ-bench, established policy-constrained tool-agent-user interaction and final-state evaluation [2]. The v0.2.0 release supplies a fixed public artifact and native trajectory files [3]. ToolSandbox adds implicit state dependencies and intermediate milestones [4]; AppWorld supplies controllable applications and programmatic state checks [5]; API-Bank studies API planning and execution [6]; AgentBench measures interactive competence [7]; and ToolLLM evaluates large-scale API selection and invocation [8]. Together, these benchmarks show that valid function syntax does not guarantee reliable execution after a timeout, repeated delivery, interruption, reordered proposal, or unobserved state change.

Agent-memory research mainly asks what information should remain available to a model. LongMemEval and very-long-term conversational-memory evaluations test retrieval across extended histories [9], [10]. MemGPT separates context and storage tiers [11], Agent Workflow Memory retrieves reusable routines [12], and Generative Agents maintain experience streams and reflections [13]. These mechanisms improve continuity and planning, but a narrative does not by itself identify which side effects committed. A transcript does not automatically bind a retry to the original operation, establish an authoritative version, or prove that external state agrees with the remembered state.

Event sourcing provides a different abstraction: an append-only event sequence acts as execution provenance from which projections can be reconstructed [14]. Empirical work documents its audit and evolution properties [15], while Microsoft and AWS architecture patterns describe replay-derived state [16], [17]. These ideas connect to causal ordering [18], linearizability [19], state-machine services [20], coordination avoidance [21], and distributed snapshots [22]. The present work does not claim full distributed serializability; it applies these principles as deterministic guards around benchmark tool execution.

Reliable recovery also requires operation semantics. HTTP defines idempotent methods, but arbitrary business actions require an application-level identity [23]. Safe retry practice therefore pairs a stable request identifier with a semantic-equivalence check [24]. Sagas address compensation for long-lived distributed work [25], while rollback-recovery research distinguishes restoration, replay, and externally visible effects [26]. Runtime verification further shows how execution events can be checked against temporal properties as they occur [27], [28], [29].

Escalation completes the control boundary. Human-AI guidance emphasizes visible system status and recoverable failure [30]. Learning-to-defer research models when automation should transfer a decision to an expert [31], including calibration against expected expert correctness [32]. NIST's AI Risk Management Framework and Generative AI Profile emphasize traceability, monitoring, and risk-appropriate oversight [33], [34]. In this experiment, escalation is narrower: a controller records a handoff decision only when the environment hash matches the failed step's expected precondition. No subsequent human action or resolution is observed.

Adjacent Evidence Across Enterprise Reliability Domains

Recent memory systems broaden the continuity side of the semantic/execution separation. Confidence-gated writing and time-aware retrieval address what should be retained across sessions [35]; federated topic-preference learning adds privacy constraints [36]; counseling copilots retrieve evidence from multi-turn dialogue [37]; and behavior-retrieval systems reuse prior interactions for interpretable personalization [38]. These approaches support compact semantic projections, but remembered statements are not, by themselves, authoritative records of committed side effects.

Evidence-grounded retrieval work makes a parallel distinction between fluent synthesis and inspectable support. Cloud-native DevOps question answering ties responses to private operational documents [39], word-level attribution exposes unsupported spans [40], bilingual incident triage links summaries to operational evidence [41], and calibrated abstention treats insufficient retrieval coverage as a reason not to answer [42]. For an action-taking agent, execution events play an analogous role: they must connect a proposed action to the state that authorized it and to the result that actually occurred.

Provenance-oriented RAG further decomposes reliability into retrieval, ranking, attribution, and deterministic explanation. Evidence-chain explanations emphasize

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

inspectable source paths [43]; budgeted multi-hop retrieval treats evidence acquisition as a controlled policy [44]; retrieval-quality prediction anticipates weak evidence before generation [45]; and findable-then-explainable code intelligence combines retrieval with concise summaries [46]. These decompositions motivate a controller that records each tool boundary rather than collapsing the workflow into one narrative memory.

AIOps studies expose the same state-and-provenance problem in operational form. Multi-source root-cause attribution joins CPU and network signals [47]; retrieved normal prototypes anchor OpenStack anomaly explanations [48]; HDFS systems connect anomaly decisions to deterministic failure narratives [49]; and DevOps copilots combine runbook retrieval with command-safety checks [50]. Such systems benefit when the evidence used to justify a recommendation is kept separate from the transaction record governing whether the recommendation may be executed.

Adversarial operation adds policy drift and deliberate manipulation. Continual red-teaming addresses newly observed jailbreaks [51], while privacy and data-integrity risk cards make prompt-injection consequences visible to human overseers [52]. Evidence-based self-verification seeks to preserve useful responses without relaxing safety boundaries [53], and trajectory-reliability models can forecast tool-use failure from partial traces [54]. Behavior-level refusal adds a further control layer [55]. In a tool workflow, these safeguards are stronger when refusal, verification, and policy updates are themselves recorded as ordered controller events.

Capacity-management research supplies another operational analogy because a forecast becomes consequential only when it drives admission, placement, or oversubscription. Calibrated time-series foundation models quantify heterogeneous GPU demand [56]; conformal envelopes bound oversubscription risk [57]; cross-cloud transfer studies workload and topology shift [58]; and multi-horizon forecasts express operational risk over several planning windows [59]. A controller therefore needs to retain not only the forecast text but also its model version, uncertainty envelope, decision threshold, and downstream state change.

Serving studies extend this requirement to sparse history and bursty demand. Few-shot forecasting targets new inference tenants [60], token-burst analysis models request and failure risk in LLM services [61], profit-aware admission selects workloads under asymmetric costs [62], and power-aware inventory planning links job forecasts to infrastructure explanations [63]. These works suggest that retries and rollbacks should be parameterized by operation semantics and risk class rather than by conversational similarity alone.

Calibration and selective refusal are also central in high-stakes classification. Default-risk tiering combines calibrated probabilities with uncertainty-driven rejection [64]; model-risk workflows join calibration, distribution-free uncertainty, and explanations [65]; fair credit scoring uses counterfactual explanations alongside group-level checks [66]; and recruiter screening applies selective refusal to uncertain resume-job matches [67]. The common design principle is to expose when the system lacks sufficient confidence, while the present study adds the requirement that the handoff occur at a verifiable environment boundary.

The same separation appears in multimodal decision support. Uncertainty-aware 3D fusion calibrates modality confidence before combining evidence [68]; medical vision-language classification treats uncertainty as part of the prediction output [69]; camera-LiDAR tracking couples probabilistic association with state estimation [70]; and lightweight medical foundation models study cross-modal transfer under limited supervision [71]. None of these tasks is identical to dual-control tool use, but each illustrates why a compact explanation should not replace the underlying state and uncertainty record.

Human oversight requires an interface layer that preserves this distinction. Scientific-search evidence cards emphasize ranking transparency and visual evidence hierarchy [72]; medical explanation cards surface uncertainty around image-based decisions [73]; patient-facing safety cards translate calibrated risk into understandable guidance [74]; and incident visualization cards organize distributed-log evidence for SRE review [75]. An escalation packet for an enterprise agent should similarly expose the committed prefix, pending operation, expected and observed versions, evidence links, and permissible recovery actions.

Recent work also treats reliability prediction and routing as pre-execution controls. Generalist-agent trajectory models forecast tool-use failure from partial traces [54]; cost-aware AIOps routers allocate requests across parsing, detection, diagnosis, and summarization stages [76]; hybrid-cloud deployment research makes placement and cost part of LLM service design [77]; and VM degradation models identify noisy-neighbor risk [78]. GPU-memory prediction likewise estimates whether a deep-learning task can be admitted safely [79]. These mechanisms are complementary to event sourcing: predictors estimate risk, whereas the controller enforces identity, order, and state consistency after a decision is made.

Taken together, the adjacent literature contributes semantic memory, retrieval provenance, calibrated uncertainty, safety verification, operational forecasting, and human-facing explanations. The unresolved systems question is how to bind those capabilities to mutable shared state so that retries, duplicate delivery, interruption, conflict, rollback, and handoff remain auditable. The following experiment isolates that question by holding the valid action plan constant and comparing complete memory-and-control packages.

The study asks three questions. First, how do four complete memory-and-control packages compare under an identical valid action plan? Second, which package components account for duplicate suppression, state-conflict response,

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

and rollback? Third, can persistent failures stop at a verifiable boundary? The evidence has three distinct layers: an audit of all three domains, an analysis of 12 primary model-result files, and deterministic controller replay over all telecom tasks. Keeping these layers separate prevents the controller percentages from being interpreted as language-model leaderboard scores.

B. METHODS

Artifact Identity And Corpus

The evaluated artifact was the τ²-bench v0.2.0 release dated October 6, 2025, at commit f8de30c298689cbe0117d76a378e7315a17e5bd8 [3]. The archive SHA-256 was 715e3dc50f03aaf9860f740a297845d61e710ecf800d7ff039d6bcf19a2f5e34. Although pyproject.toml carries the development string 0.2.1-dev, the release tag, notes, commit, archive name, and content hash jointly fix the evaluated artifact. Table 1 records the corpus identity used by every subsequent analysis.

Table 1 Artifact Identity And Evaluated Telecom Corpus

Item	Value
Release tag and date	v0.2.0; October 6, 2025
Release commit	f8de30c298689cbe0117d76a378e7315a17e5bd8
Archive SHA-256	715e3dc50f03aaf9860f740a297845d61e710ecf800d7ff039d6bcf19a2f5e34
Telecom tasks	114
Listed telecom actions	516
User / assistant actions	393 / 123
Mutating actions	496
Environment assertions	209
Primary result files / trajectories	12 / 4,448

The domain audit covers 278 tasks and 1,217 listed actions (Table 2 and Figure 1). Airline and retail contain assistant-side listed actions, whereas telecom contains 393 user-side actions and 123 assistant-side actions. Controlled replay therefore focuses on telecom because its listed plan explicitly exercises both actors. The primary model-result analysis retains all three domains and is not restricted to telecom.

Table 2 Domain Task And Action Profile

Domain	Tasks	Actions	Assistant	User	Assertions
Airline	50	148	148	0	0
Retail	114	553	553	0	0
Telecom	114	516	123	393	209
Total	278	1,217	824	393	209

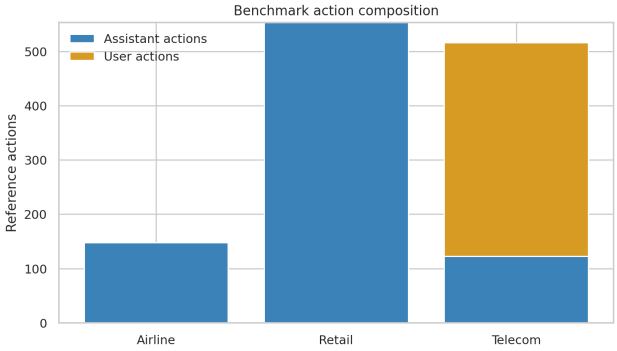

Figure 1 Reference-Action Composition Across The Three Domains. Telecom Is The Only Profiled Domain With Listed User-Side Actions.

Oracle-Plan Isolation And Native Execution

Each telecom task's evaluation criteria supply one valid sequence of tool calls used to construct the target environment. The experiment treats that sequence as an oracle plan to isolate execution behavior after semantic planning. It does not score trajectory similarity and does not assume that the listed sequence is the only valid solution. Every controller receives the same requestor, tool name, and arguments, so observed differences arise from the configured controller package rather than from a different plan.

For each task, the native environment is initialized from task-specific data and initialization actions. The listed sequence is replayed once to obtain expected pre-action hashes, post-action hashes, and the combined target hash of the assistant and user databases. Tool synchronization follows every call. All 114 plans complete, and all 209 environment assertions pass. Each experimental row then begins from a separately initialized environment; no mutated state is shared between rows. The verifier performs 27,260 individual assertion invocations in the main campaign and 23,480 in the component campaign, for 50,740 total.

The four controller configurations intentionally couple prompt representation with control capabilities. Full History, for instance, could in principle be paired with a version gate, but the evaluated Full History package is not. Results therefore compare named packages, not the isolated causal effect of text length, event storage, or any single memory format. Component evidence comes from the separate event-package ablations.

Controller-Policy Packages

Stateless exposes the current action and latest result to its decision rule. It has no retry policy, retained resume position, duplicate guard, state gate, rollback, or escalation decision, and it does not retain a complete event sequence. Full History exposes the accumulated ordered transcript. It retries a pre-commit transient fault, resumes from its retained sequence position after a boundary interruption, and suppresses a duplicate when the exact canonical action signature appears in history. It does not compare the

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

transcript with an external version and accepts the injected adjacent order reversal.

Summary Memory exposes a next-step counter plus the three most recent signatures and statuses. It retries when canonical arguments contain no more than 14 lexical tokens, recognizes duplicates within that window, and applies an adjacent tool-name heuristic: different names are returned to canonical order. This is not a verified dependency graph or policy-level partial order. Summary Memory preserves its sequence counter across the boundary interruption and records an escalation decision when its retry rule declines a call.

Event-Sourced Memory retains a workflow-lifetime append-only journal outside prompt exposure. Each event contains the step, canonical signature, deterministic idempotency identity, status, pre-hash, post-hash, and bounded tool result. Its prompt projection contains the next step, shortened version, pending key, and two recent statuses. The event package checks committed identity for duplicate delivery and applies the adjacent name gate plus a configured state-precondition heuristic for order repair. For the stale intervention it recognizes an effective out-of-band mutation, reinitializes the environment, and executes the canonical sequence. Figure 2 summarizes the design boundary.

Table 3 Evaluated Controller-Policy Packages

Package	Prompt exposure	Retry / resume	Duplicate / order	Version / rollback / escalation
Stateless	Current action and result	None / restart	None / accepts	None / none / none
Full History	Accumulated transcript	Yes / retained boundary	Global signature / accepts	None / none / yes
Summary Memory	Next step and last three	Conditional / retained boundary	Three-event window / name heuristic	None / none / yes
Event-Sourced	Compact projection; full journal external	Yes / retained boundary	Global identity / name-plus-state heuristic	Configured / reinitialize / yes

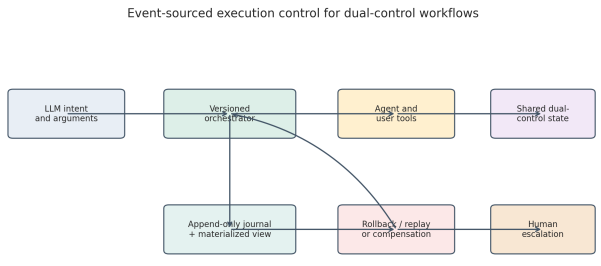

Figure 2 Event-Sourced Execution Control For Dual-Control Workflows. The Compact Projection Enters Model Context; The Event Journal Remains In The Controller Layer.

Deterministic Interventions And Coverage

Five recoverable interventions and one persistent-failure condition are applied. A SHA-256 digest of task identifier, condition, and trial selects the intervention position, pairing the four packages on the same task and position. Each applicable task has five rows per package. Clean and interruption cover all 114 tasks. Conditions that select a user-side mutation cover 94 tasks.

Transient failure replaces a selected mutating call with a pre-commit fault. Duplicate call redelivers one planned user mutation after the canonical sequence. Order swap reverses a selected user action and its adjacent predecessor; the experiment does not infer whether those operations truly depend on one another. Interruption clears or preserves controller-facing progress at an action boundary while the same process and environment remain active. It does not serialize, reload, or test a mid-write crash. Stale state pre-applies one planned user mutation before step zero. In the persistent condition the fault remains active for any permitted retry; a package therefore observes one or two pre-commit errors according to its retry rule before the stop-and-escalate decision.

The main campaign contains 13,960 rows: 2,280 clean, 9,800 recoverable-fault, and 1,880 persistent-failure rows. The component campaign adds 12,080 rows. Table 4 gives the exact paired coverage.

Table 4 Intervention Protocol And Paired Run Counts

Condition	Tasks/package	Trials/task	Main rows	Ablation rows
Clean	114	5	2,280	2,280
Transient failure	94	5	1,880	1,880
Duplicate call	94	5	1,880	1,880
Order swap	94	5	1,880	1,880
Boundary interruption	114	5	2,280	2,280
Pre-applied user mutation	94	5	1,880	1,880
Persistent failure	94	5	1,880	—
Total	—	—	13,960	12,080

Outcomes And Statistical Analysis

Task success requires exact agreement with the combined assistant/user target hash, passage of the task's assertion conjunction, and autonomous completion without an escalation flag. Safe resolution requires either successful completion without a controller-induced violation or a configured escalation decision at the expected boundary. In the main persistent condition, a correct escalation decision means that the current combined hash equals the listed pre-action hash for the failed step. In the No Rollback stale-state branch, it means that the controller stops without further mutation and verifies preservation of the observed post-mutation hash. The experiment does not invoke a person, measure response quality, or observe later resolution.

A violation denotes execution of a recognized duplicate, continuation after an unhandled pre-commit fault, restart of completed work, acceptance of the injected adjacent

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

reversal, or continuation through the pre-applied mutation. An unresolved failure is a run that neither completes nor reaches a state-verified escalation decision. Retry, rollback, suppression, repair, conflict-detection, tool-error, and executed-call counts are stored per row.

The memory-token measure is a prompt-exposure proxy. A deterministic lexer counts alphanumeric spans and punctuation in each controller snapshot and sums them across the run. It is not a model tokenizer, resident-memory measurement, journal-size measurement, or durable-storage cost. Local wall latency includes Python orchestration and native tool calls but excludes any model, network, or external event-store latency.

For paired inference, recoverable outcomes are first averaged within each task and package. Event-Sourced is compared with each baseline through 10,000 task-cluster bootstrap resamples and one-sided paired Wilcoxon tests; Holm correction controls the three tests. These comparisons estimate the effect of the complete package assignments. The ablation campaign gives more focused—but still package-conditional—evidence about global idempotency, precondition policy, and rollback.

Primary Model-Trajectory Analysis

The release supplies 12 primary result files: four model configurations across airline, retail, and telecom. Each model contributes four trials for 278 tasks, or 1,112 trajectories; the combined collection contains 4,448 trajectories. The packaged GPT-4.1 mini filenames use the release's base mode, while the other three model groups use default mode. The analysis is therefore an audit of bundled trajectories, not a controlled model rerun. Native success is the stored benchmark reward. For a task with c successful runs among n=4, Pass@k is the order-invariant estimate 1-C(4-c,k)/C(4,k), and strict pass^k is C(c,k)/C(4,k). The aggregate values in Table 6 and Figure 3 are read directly from bundled_trajectory_metrics.csv.

C. RESULTS AND DISCUSSION

Validation And Primary Trajectory Context

All 114 native oracle-plan replays completed, all 516 listed telecom actions executed, and all 209 assertions passed. In the main campaign, 2,280/2,280 clean rows reached the exact state, 2,280/2,280 passed the assertion conjunction, and 2,280/2,280 met task success. Thus every package executes the same plan correctly without an intervention.

Across the 4,448 primary model trajectories, native success is 2,528/4,448, or 56.83%. Table 5 shows that Claude 3.7 Sonnet has the highest pooled rate and GPT-4.1 mini the lowest. Each row pools three domain files and all 278 tasks for one model. These outcomes include language-model dialogue and planning, unlike the oracle-plan controller replay.

Table 5 Native Reward Success Pooled Across The Primary Result Files

Model	Files	Tasks	Trajectories	Successes	Success rate
Claude 3.7 Sonnet	3	278	1,112	684	61.51%
GPT-4.1	3	278	1,112	606	54.50%
GPT-4.1 mini	3	278	1,112	602	54.14%
o4-mini	3	278	1,112	636	57.19%
Pooled	12	1,112	4,448	2,528	56.83%

Order-invariant repeated-trial reliability across all three domains is reported in Table 6. Pass@4 reaches 79.14% for Claude 3.7 Sonnet, whereas strict pass^4 ranges from 27.70% to 41.37%. Figure 3 traces strict pass^k across k=1 through 4. The separation between best-of-k and all-of-k reliability shows why a single successful trajectory does not establish dependable repeated operation.

Table 6 Order-Invariant Repeated-Trial Reliability Across All Three Domains

Model	Traj.	Success	Pass@1	Strict p^1	Pass@4	Strict p^4
Claude 3.7 Sonnet	1,112	61.51%	61.51%	61.51%	79.14%	41.37%
GPT-4.1	1,112	54.50%	54.50%	54.50%	71.58%	36.69%
GPT-4.1 mini	1,112	54.14%	54.14%	54.14%	75.18%	27.70%
o4-mini	1,112	57.19%	57.19%	57.19%	75.54%	36.33%

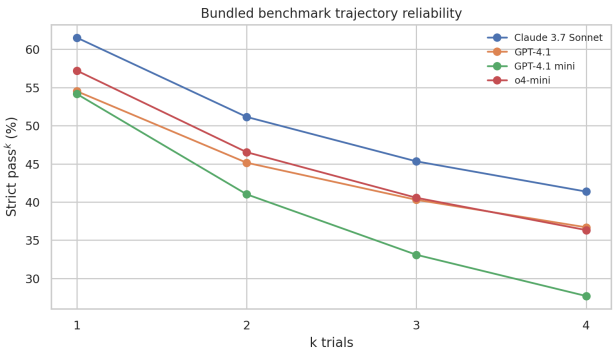

Figure 3 Order-Invariant Strict Pass^K Over The Primary Trajectories Pooled Across Airline, Retail, And Telecom.

The primary results and controller results answer different questions. Primary rewards reflect model reasoning, conversation, tool selection, and execution. Controller success measures whether a fixed listed action sequence reaches the target under deterministic interventions. Event-Sourced's complete recoverable score is therefore conditional on a valid plan and configured repair rules; it is not a native model score.

Pooled Recovery And Exact-State Behavior

Across 2,450 recoverable rows per package, Event-Sourced reaches 100.00% task success and safe resolution (Table 7). Full History reaches 89.55% task success, Summary Memory 80.37%, and Stateless 45.92%. Event-Sourced is the only package with no unresolved failures and no logged violations.

The gap between task success and safe resolution is substantive. A run may reach the target after accepting an

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

injected order reversal or repeating an operation whose second execution happens to be a no-op. Final equality can therefore coexist with a process violation. Full History's exact-state rate is materially higher than its safe-resolution rate because the package accepts every adjacent order reversal and lacks a pre-applied-mutation guard.

Table 7 Pooled Recoverable-Fault Outcomes

Package	Runs	Task success	Exact state	Safe resolution	Unresolved failure	Violation
Stateless	2,450	45.92 %	45.92 %	1.51%	54.08%	98.49 %
Full History	2,450	89.55 %	89.55 %	63.14 %	10.45%	36.86 %
Summary Memory	2,450	80.37 %	80.37 %	73.71 %	13.35%	26.29 %
Event-Sourced	2,450	100.00%	100.00%	100.00 %	0.00%	0.00%

Condition-level success in Table 8 and Figure 4 reveals the operative differences. Stateless almost always omits the failed mutation, whereas Full History and Event-Sourced retry every transient case. Summary Memory automatically completes the calls that meet its lexical retry rule and records a state-verified escalation decision in the remaining cases. Delayed duplicate delivery separates global history or idempotency identity from the bounded recent window. Boundary interruption separates a retained sequence position from restart behavior.

Table 8 Task Success By Recoverable Intervention

Package	Transient	Duplicate	Order swap	Boundary reset	Pre-applied mutation
Stateless	1.06%	44.89%	92.77%	39.47%	52.77%
Full History	100.00 %	100.00 %	92.77%	100.00 %	52.77%
Summary Memory	67.23%	77.66%	100.00 %	100.00 %	52.77%
Event-Sourced	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %

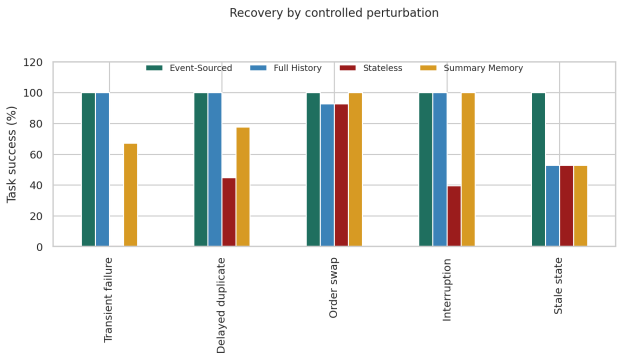

Figure 4 Task Success Under The Five Recoverable Interventions. Each Bar Aggregates Five Paired Rows Over All Applicable Tasks.

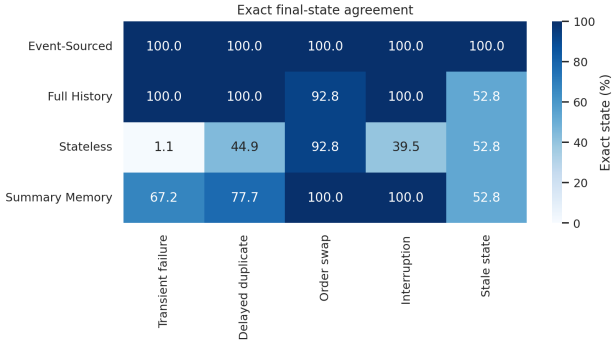

Figure 5 Exact Final-State Agreement By Package And Recoverable Intervention.

Transient failure yields 1.06% Stateless success. Full History and Event-Sourced complete all rows. Summary Memory completes 67.23%; its other 154 rows end in a configured escalation decision at the listed precondition, so transient safe resolution is 100.00%.

For duplicate delivery, Full History and Event-Sourced suppress every replay. Summary Memory reaches 77.66%, while Stateless reaches 44.89%. Some replayed writes leave the final hash unchanged, but an accepted replay is still marked as a violation. This distinction is visible when Table 7 is compared with the exact-state heatmap in Figure 5.

For order swaps, Stateless and Full History reach 92.77% exact state because some adjacent operations commute. Both packages nevertheless accept every injected reversal. Summary Memory restores canonical order when adjacent tool names differ and reaches 100.00% exact state, but its safe-resolution rate is 95.11%. Event-Sourced additionally treats same-name pairs as conflicts through its configured precondition flag. These repairs are heuristic package rules, not evidence that the code proves a dependency graph.

At the boundary interruption, Full History, Summary Memory, and Event-Sourced continue from the retained midpoint and each reach complete task success. Stateless clears its progress and restarts, reaching 39.47%. Because the same in-memory process, ledger object, and environment remain active, this result supports completed-step boundary resume only; it does not establish crash recovery, process reload, checkpoint durability, or response-loss handling.

The pre-applied-mutation condition is the clearest package separation. One planned user write is executed once before step zero. In 433 of 470 Event-Sourced rows, that action changes the combined hash; in the other 37 it is ineffective under the initialized state. Stateless, Full History, and Summary Memory each reach 52.77%. Event-Sourced reinitializes the effective-change rows and then runs the canonical sequence, reaching complete success. This tests a duplicate out-of-band planned mutation plus full reinitialization, not general reconciliation of an independent concurrent update.

Safety Operations And Escalation Decisions

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Table 9 separates recovery from the control actions associated with it. Event-Sourced performs retries only in transient rows, global suppressions only in duplicate rows, and reinitialization only when the pre-applied mutation changes state. Its conflict count also includes same-name order cases handled through the precondition flag. Full History matches the retry and global-signature suppression rates but has no configured rollback or external-conflict rule. Summary Memory applies fewer retries and suppressions because its prompt-visible rules are bounded. Figure 6 places recovery and violation on the same operating plane.

Table 9 Recovery, Violations, And Control-Operation Means

Package	Recovery	Violation	Retries/run	Rollbacks/run	Suppressions/run	Conflicts/run
Stateless	45.92%	98.49%	0.000	0.000	0.000	0.000
Full History	89.55%	36.86%	0.192	0.000	0.192	0.000
Summary Memory	80.37%	26.29%	0.129	0.000	0.115	0.000
Event-Sourced	100.00%	0.00%	0.192	0.177	0.192	0.186

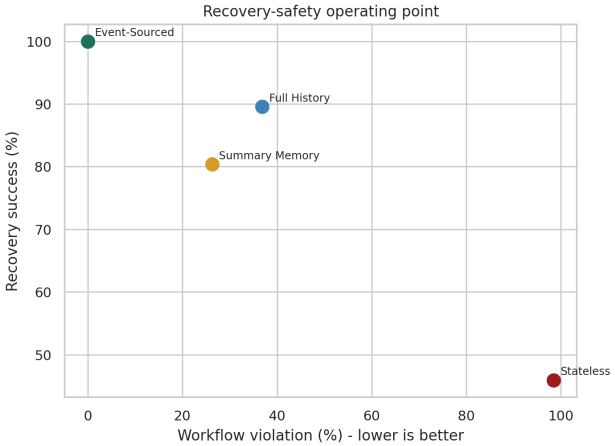

Figure 6 Recovery-Success And Controller-Induced-Violation Operating Points Over Recoverable Interventions.

Persistent failure tests the decision to stop at a checkable boundary. Event-Sourced, Full History, and Summary Memory each record escalation in all 470 rows, and the saved current-state hash matches the listed pre-action hash in every one of those rows (Table 10). Autonomous task success is zero because the workflow intentionally stops. The row flag establishes a state-verified escalation decision only. It does not call a transfer tool, contact a person, or evaluate what happens after the decision.

Stateless records no escalation decision and continues after the missing operation. Its autonomous task-success rate is 0.43%, but every row violates the failure policy and safe resolution is zero. The stateful packages execute a mean committed prefix of 2.27 calls before stopping; Stateless executes 4.28 calls on average because it continues into the suffix.

Table 10 Persistent-Failure Escalation Decisions

Package	Cases	Auto success	Escalation	State-verified	Safe resolution	Mean executed
Stateless	470	0.43%	0.00%	0.00%	0.00%	4.28
Full History	470	0.00%	100.00%	100.00%	100.00%	2.27
Summary Memory	470	0.00%	100.00%	100.00%	100.00%	2.27
Event-Sourced	470	0.00%	100.00%	100.00%	100.00%	2.27

A production escalation packet should include operation identity, committed prefix, expected and observed versions, and the error sequence. Learning-to-defer methods could estimate when escalation is preferable [31], [32], while the execution controller would still verify the boundary. This division aligns traceable intervention with human-AI guidance and risk-management practice [30], [33], [34] without assuming that escalation itself resolves the task.

Prompt Exposure And Local Execution Cost

Event-Sourced has a mean prompt-exposure proxy of 219.15 tokens per recoverable row, 89.73% lower than Full History's 2,134.61 (Table 11). Its median peak is 37, compared with 516 for Full History. Figure 7 shows the reliability-exposure relationship. The measurement supports a compact prompt projection; it is not a storage-size result. Full History retains its transcript, Event-Sourced retains its journal, Summary Memory retains only its bounded recent window and progress state, and Stateless retains no decision journal.

Event-Sourced median local wall latency is 2.644 ms and P95 is 6.189 ms. Summary Memory has the lowest median at 2.409 ms. These sub-10-ms values contain no model inference, network call, or external journal write and should not be extrapolated to production response time.

Table 11 Prompt-Exposure Proxy And Local Latency

Package	Mean proxy	Median peak	Median ms	P95 ms
Stateless	395.93	102	2.446	5.281
Full History	2,134.61	516	2.597	5.695
Summary Memory	505.69	126	2.409	5.016
Event-Sourced	219.15	37	2.644	6.189

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

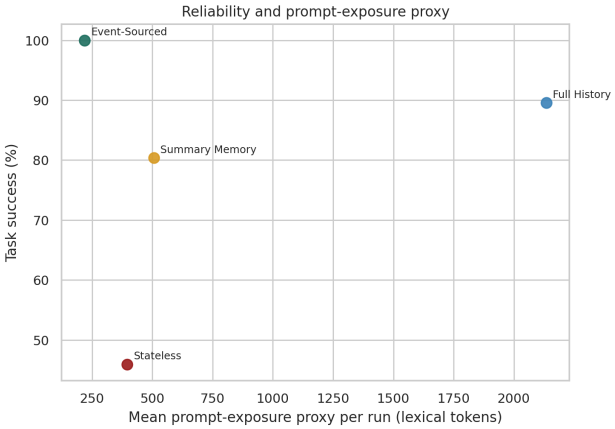

Figure 7 Recoverable Success Versus Mean Prompt-Exposure Proxy. Bubble Area Encodes Median Local Wall Latency.

Component Packages And Paired Inference

Table 12 and Figure 8 show that the complete event package outperforms each component removal. The No Durable Idempotency Index configuration reaches 94.04%. It removes the global committed-identity lookup but retains duplicate visibility within the two-event recent window, so it is not a removal of every duplicate check. No Preconditions reaches 90.94% and loses detection of the effective pre-applied mutation. No Rollback reaches 82.33% autonomous success but 100.00% safe resolution because its configured policy preserves the observed changed-state hash and stops for handoff rather than completing the canonical plan.

The No Rollback safety value requires particular care. It verifies non-overwrite of the observed user-mutated state at the handoff boundary; it does not demonstrate restoration of the original state or a later human resolution. The ablation therefore shows that rollback raises autonomous completion, while state-preserving handoff supplies the evaluated fallback when restoration is disabled.

Table 12 Event-Package Component Analysis

Configuration	Runs	Task success	Exact state	Safe resolution	Violation
No Rollback	2,450	82.33%	82.33%	100.00%	0.00%
No Preconditions	2,450	90.94%	90.94%	81.39%	18.61%
No Durable Index	2,450	94.04%	94.04%	88.78%	11.22%
Event-Sourced	2,450	100.00%	100.00%	100.00%	0.00%

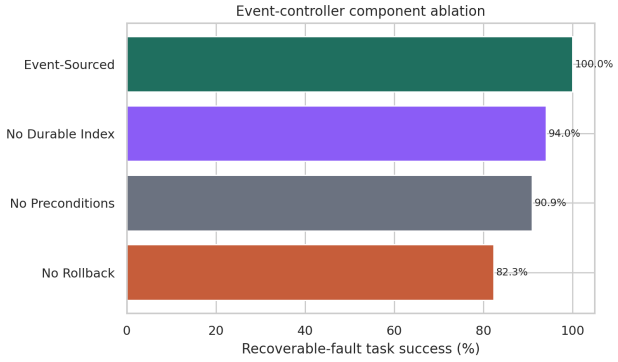

Figure 8 Recoverable-Fault Autonomous Task Success For The Complete Event Package And Three Component Removals.

Task-clustered inference in Table 13 reduces the influence of different condition eligibility. Event-Sourced exceeds Stateless by 46.49 percentage points, Full History by 8.98 points, and Summary Memory by 16.88 points. Every 95% bootstrap interval excludes zero, and all Holm-adjusted paired tests remain significant. These are comparisons among complete deterministic package assignments under the oracle plan; they do not isolate a universal effect of event storage.

Table 13 Task-Clustered Comparisons With Event-Sourced

Comparison	Tasks	Difference	95% bootstrap CI	Raw p	Holm p
Event-Sourced vs. Stateless	114	46.49 pp	[41.44, 51.40] pp	1.75e-17	5.26e-17
Event-Sourced vs. Full History	114	8.98 pp	[7.61, 10.39] pp	1.86e-15	1.86e-15
Event-Sourced vs. Summary Memory	114	16.88 pp	[14.70, 19.09] pp	3.43e-17	6.87e-17

Workflow Length, Boundaries, And Operational Implications

The package ordering persists across all workflow-length groups (Table 14). Event-Sourced remains at complete success from one through twelve listed actions. Stateless falls from 55.43% in the shortest group to 39.43% in the longest. Full History remains between 87.44% and 91.20%, while Summary Memory ranges from 77.56% to 84.91%. The event result is not explained by one unusually difficult length bin.

Table 14 Recoverable Success By Listed Workflow Length

Package	1–3 (n=875)	4–6 (n=900)	7–9 (n=500)	10–12 (n=175)
Stateless	55.43%	40.22%	41.80%	39.43%
Full History	91.20%	87.44%	90.20%	90.29%
Summary Memory	84.91%	77.56%	78.40%	77.71%
Event-Sourced	100.00%	100.00%	100.00%	100.00%

The evidence supports a two-plane architecture. Semantic context should retain user intent, policy, and facts required for reasoning. Execution state should retain operation identity, order, observed outcomes, and recovery decisions. Full History demonstrates that narrative completeness can

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

support retry and boundary resume, but it cannot account for an event outside its transcript. Summary Memory demonstrates that a compact progress pointer can support ordinary continuation while losing older identity. The event package succeeds because its configured journal and guards jointly align the planned next step with execution history.

Production use requires stronger mechanisms than the benchmark harness. Every mutating service should publish an idempotency contract, precondition, postcondition, retry class, and compensation behavior. Irreversible effects such as payments, messages, and dispatch require application-level idempotency or Saga compensation [23], [24], [25], [26]. A real event store would also need persistence, schema evolution, access control, replay isolation, and monitoring. None of those storage-system properties is measured by the in-memory controller object.

The intervention campaign contains one controlled transformation per row. Transient errors occur before commit; ambiguous after-commit timeouts and response loss are not exercised. The boundary interruption does not terminate a process. The pre-applied user mutation is selected from the listed plan rather than drawn from an independent concurrent actor. Order repair uses adjacent names and a configured state flag rather than a dependency proof. These choices make causal comparison tractable but narrow the external validity of the resulting percentages.

The escalation evidence is similarly bounded. It verifies a controller decision and, in the main persistent condition, checks the current hash against the expected pre-action state. It does not measure a person's response, time to resolution, correction quality, or workload. A deployment should treat escalation as the beginning of another audited workflow and retain the same event identity and state evidence through that process.

Within those boundaries, the study offers a concrete engineering rule: do not use a conversation transcript as the transaction log. The model can propose intent and arguments, while an orchestration layer assigns identity, checks the next boundary, records the outcome, and exposes a compact projection for the next turn. Runtime monitors can then distinguish detected, repaired, escalated, and unresolved states [27], [28], [29], while organizational controls determine which operations require human review [30], [33], [34].

Cross-Domain Deployment And Oversight Implications

A production event journal should retain evidence at the granularity required for later forensic review. LogBERT-style anomaly detection can identify unusual event windows [80]; CloudTrail sequence models can place events within an interpretable attack chain [81]; conformal alert control can attach calibrated alert status and an evidence-grounded ticket [82]; SSD-health modeling illustrates the same need for versioned predictive maintenance evidence [83]; and host-based intrusion systems can localize system-call windows before generating a narrative [84]. For these applications, an escalation record should preserve the exact evidence window, detector and threshold version, operation identity, current state hash, and the reason automatic recovery stopped.

The state-consistency requirement extends beyond cloud incidents to nonstationary planning. Cross-dataset parcel priors transfer demand structure into last-mile capacity estimates [85], while probabilistic bike-sharing forecasts explicitly model changing weather and seasonal regimes [86]. Combined with the GPU and VM studies discussed above, these examples imply that replay should restore both the business state and the decision context: the data cutoff, feature projection, forecast distribution, and policy threshold that produced the original action.

High-stakes detection studies reinforce the need to distinguish model output from control authority. Extreme-imbalance fraud detection evaluates cost-sensitive and one-class alternatives [87]; TinyLLM-assisted IoT detection targets real-time network constraints [88]; bankruptcy prediction combines imbalance handling with explainable financial portraits [89]; language-guided intrusion detection narrows the feature set for DDoS analysis [90]; and deep-autoencoder traffic models classify IoT behavior and anomalies [91]. An enterprise controller can consume these scores, but it should record the score as evidence and apply a separately versioned action policy.

Predictive DDoS mitigation illustrates why controller events must also capture external effects: mitigation changes routing, filtering, and resource allocation, not merely a label [92]. The same principle applies to any security action that can block users, isolate hosts, rotate credentials, or modify infrastructure. Idempotency keys prevent repeated effects, preconditions protect against stale observations, and compensation or verified handoff is required when an action cannot be safely reversed.

Advertising and recommendation research offers concrete examples of policy evolution. Privacy-robust uplift modeling estimates incremental response without persistent identifiers [93]; DriftGuard couples multi-signal drift detection with safe retraining or rollback [94]; LLM-assisted uplift modeling turns feature interactions into audience-creative-channel policies [95]; and conservative off-policy evaluation selects slot policies from logged bandit data [96]. Event-sourced deployment would attach model, cohort, treatment, propensity, and policy versions to each decision so that a rollback restores a coherent policy state rather than only an earlier text prompt.

Budgeted auto-bidding and counterfactual ranking make the same requirement explicit because a single action consumes scarce resources. Distributional constrained reinforcement learning enforces budget and risk limits in first-price auctions [97], privacy-safe marketing-mix optimization operates after identifier loss [98], and counterfactual learning-to-rank evaluates ad policies from logged interactions [99]. A durable controller should therefore record the budget snapshot, spend commitment, observed

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

outcome, and any compensating adjustment, preventing an interrupted conversation from silently resetting fiscal state.

Evidence-constrained agents are increasingly proposed for regulated financial and legal workflows. Accounting-aware retrieval supports due diligence of tokenized trade receivables [100]; multi-regulation RAG structures cross-border product counsel [101]; evidence-constrained agents monitor disclosure, settlement, and secondary-market risk [102]; and ontology enrichment organizes FinTech M&A information from SEC disclosures [103]. These systems need a provenance chain that distinguishes retrieved authority, model interpretation, approved action, and the resulting ledger or disclosure state.

Human review in those domains also depends on presentation. Offline reranking improves the ordering of evidence in financial search [104]; numerical-reasoning guardrails check a quant assistant against SEC and FRED evidence [105]; accounting disclosure cards make source-grounded explanations reviewable [106]; and financial-risk dashboards study hierarchy, chart comprehension, and explainability [107]. The event-sourced handoff envisioned here is compatible with these interfaces because it supplies a stable operation identity and state snapshot that a card or dashboard can display without inventing provenance.

Recommendation systems show how semantic personalization and transactional state can diverge. Explainable e-commerce cards communicate recommendation evidence and confidence [108]; self-supervised customer representations support segmentation and next-purchase prediction [109]; review-grounded recommendation evaluates whether explanations remain faithful to product evidence [110]; and machine-learning product classification organizes items for personalized retrieval [111]. A purchase, subscription, or inventory mutation nevertheless requires a separate execution ledger even when the recommendation rationale is well grounded.

Education and counseling provide a parallel human-in-the-loop setting. Rubric-grounded essay scoring emphasizes auditability and formative feedback [112]; strategy-aware therapist imitation controls supportive response style [113]; student-retention analytics pair early warning with intervention rationales [114]; teacher-facing learning analytics expose grade and dropout evidence [115]; and counseling recommendation cards link suggested support to multi-turn evidence [116]. In each case, the explanation may support judgment, but intervention status, consent, assignment, and follow-up should remain in an authoritative workflow record.

Interface assurance becomes especially important when users cross languages or when the interface itself can manipulate choice. Trust-calibrated multilingual RAG targets humanitarian information access [117]; text-grounded design-rationale cards connect layout metadata to design decisions [118]; dark-pattern detection explains potentially manipulative microcopy [119]; and visual-brief cards convert creative intentions into structured design choices [120]. These systems suggest that human oversight should expose both the content rationale and the controller state that determines what action is currently allowed.

Generated and progressively rendered interfaces introduce further state transitions. Vision-language methods translate sketches into interactive web prototypes and evaluate structural consistency [121]; LLM-based design critics compare generated feedback with human judgment [122]; layout-aware PDF rendering prioritizes slices to reduce time to a functional first frame [123]; and structured visual briefs keep generated advertising concepts editable [124]. Event identity and versioning can prevent a stale critique, partial render, or outdated brief from being applied to the wrong design revision.

The need for bounded automation is not confined to language-first systems. Autonomous-driving detection research modifies visual recognition architectures for safety-sensitive perception [125]; DenseNet-based cancer-image classification illustrates high-stakes imaging decisions [126]; calibration-light motor-imagery BCI studies subject-independent transfer [127]; and AI-assisted minimal residual disease evaluation supports multiple-myeloma prediction [128]. These studies do not validate the controller percentages reported here, but they identify domains in which evidence, uncertainty, and state-consistent handoff are operationally consequential.

Across these domains, the deployable pattern is consistent: the model proposes or explains; a controller assigns a stable identity, verifies preconditions, records the observed result, and exposes only the compact state needed for the next decision. When automatic continuation is unsafe, the same journal should initialize a human workflow rather than terminate at an unstructured error message. This preserves the distinction established by the experiment between semantic memory, execution provenance, and verified escalation.

D. CONCLUSIONS

This study separates three evidence layers in the fixed τ^2-bench v0.2.0 artifact: a 278-task domain audit, 4,448 primary model trajectories across twelve files, and deterministic controller replay over all 114 telecom tasks. The controller campaign contains 13,960 main and 12,080 component-analysis runs. All clean executions validate the listed plans and native environment before any intervention is applied.

Under recoverable interventions, Event-Sourced reaches 100.00% task success and safe resolution with no logged violations. Full History reaches 89.55% task success but cannot detect the pre-applied mutation; Summary Memory reaches 80.37% while trading older identity for compact prompt exposure; Stateless reaches 45.92%. Persistent-failure results show that stateful packages can record a state-verified escalation decision, but they do not establish a later human outcome.

The principal conclusion concerns architecture rather than a leaderboard claim. Semantic memory and execution state

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

are distinct. Reliable dual-control workflows benefit from stable operation identity, explicit boundary checks, rollback or compensation, and auditable escalation evidence. The measured advantages belong to the evaluated coupled policy packages under an oracle plan and deterministic interventions. Extending them to model-planned trajectories, true process recovery, concurrent independent updates, and durable external services is the next empirical step.

E. REFERENCES

[1] V. Barres, H. Dong, S. Ray, X. Si, and K. Narasimhan, "tau2-Bench: Evaluating Conversational Agents in a Dual-Control Environment," in Proc. 43rd Int. Conf. Machine Learning, 2026, arXiv:2506.07982.

[2] S. Yao, N. Shinn, P. Razavi, and K. Narasimhan, "tau-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains," in Proc. 13th Int. Conf. Learning Representations, 2025.

[3] [3] Sierra Research, "tau2-bench v0.2.0: Web-Based Leaderboard," GitHub Release, Oct. 6, 2025.

[4] J. Lu et al., "ToolSandbox: A Stateful, Conversational, Interactive Evaluation Benchmark for LLM Tool Use Capabilities," in Findings of NAACL, pp. 1160–1183, 2025.

[5] H. Trivedi et al., "AppWorld: A Controllable World of Apps and People for Benchmarking Interactive Coding Agents," in Proc. 62nd Annual Meeting of the ACL, pp. 16022–16076, 2024.

[6] M. Li et al., "API-Bank: A Comprehensive Benchmark for Tool-Augmented LLMs," in Proc. EMNLP, pp. 3102–3116, 2023.

[7] X. Liu et al., "AgentBench: Evaluating LLMs as Agents," in Proc. 12th Int. Conf. Learning Representations, 2024.

[8] Y. Qin et al., "ToolLLM: Facilitating Large Language Models to Master 16000+ Real-World APIs," in Proc. 12th Int. Conf. Learning Representations, 2024.

[9] D. Wu, H. Wang, W. Yu, Y. Zhang, K.-W. Chang, and D. Yu, "LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory," in Proc. 13th Int. Conf. Learning Representations, 2025.

[10] A. Maharana, D.-H. Lee, S. Tulyakov, M. Bansal, F. Barbieri, and Y. Fang, "Evaluating Very Long-Term Conversational Memory of LLM Agents," in Proc. 62nd Annual Meeting of the ACL, pp. 13851–13870, 2024.

[11] C. Packer et al., "MemGPT: Towards LLMs as Operating Systems," arXiv:2310.08560, 2023.

[12] Z. Z. Wang, J. Mao, D. Fried, and G. Neubig, "Agent Workflow Memory," in Proc. 42nd Int. Conf. Machine Learning, pp. 63897–63911, 2025.

[13] J. S. Park et al., "Generative Agents: Interactive Simulacra of Human Behavior," in Proc. 36th ACM UIST, 2023.

[14] M. Fowler, "Event Sourcing," martinfowler.com, 2005.

[15] M. Overeem, M. Spoor, S. Jansen, and S. Brinkkemper, "An Empirical Characterization of Event Sourced Systems and Their Schema Evolution—Lessons from Industry," J. Systems and Software, vol. 178, Art. 110970, 2021.

[16] Microsoft, "Event Sourcing Pattern," Azure Architecture Center, 2026.

[17] Amazon Web Services, "Event Sourcing Pattern," AWS Prescriptive Guidance, 2026.

[18] L. Lamport, "Time, Clocks, and the Ordering of Events in a Distributed System," Commun. ACM, vol. 21, no. 7, pp. 558–565, 1978.

[19] M. P. Herlihy and J. M. Wing, "Linearizability: A Correctness Condition for Concurrent Objects," ACM Trans. Programming Languages and Systems, vol. 12, no. 3, pp. 463–492, 1990.

[20] F. B. Schneider, "Implementing Fault-Tolerant Services Using the State Machine Approach: A Tutorial," ACM Computing Surveys, vol. 22, no. 4, pp. 299–319, 1990.

[21] P. Bailis et al., "Coordination Avoidance in Database Systems," Proc. VLDB Endowment, vol. 8, no. 3, pp. 185–196, 2014.

[22] K. M. Chandy and L. Lamport, "Distributed Snapshots: Determining Global States of Distributed Systems," ACM Trans. Computer Systems, vol. 3, no. 1, pp. 63–75, 1985.

[23] R. Fielding, M. Nottingham, and J. Reschke, "HTTP Semantics," RFC 9110, June 2022.

[24] M. Featonby, "Making Retries Safe with Idempotent APIs," Amazon Builders' Library, Amazon Web Services.

[25] H. Garcia-Molina and K. Salem, "Sagas," in Proc. ACM SIGMOD Int. Conf. Management of Data, pp. 249–259, 1987.

[26] E. N. Elnozahy, L. Alvisi, Y.-M. Wang, and D. B. Johnson, "A Survey of Rollback-Recovery Protocols in Message-Passing Systems," ACM Computing Surveys, vol. 34, no. 3, pp. 375–408, 2002.

[27] M. Leucker and C. Schallhart, "A Brief Account of Runtime Verification," J. Logic and Algebraic Programming, vol. 78, no. 5, pp. 293–303, 2009.

[28] A. Bauer, M. Leucker, and C. Schallhart, "Runtime Verification for LTL and TLTL," ACM Trans. Software Engineering and Methodology, vol. 20, no. 4, Art. 14, 2011.

[29] K. Havelund and G. Rosu, "Monitoring Programs Using Rewriting," in Proc. 16th IEEE Int. Conf. Automated Software Engineering, pp. 135–143, 2001.

[30] S. Amershi et al., "Guidelines for Human-AI Interaction," in Proc. CHI Conf. Human Factors in Computing Systems, Art. 3, pp. 1–13, 2019.

[31] H. Mozannar and D. Sontag, "Consistent Estimators for Learning to Defer to an Expert," in Proc. 37th Int. Conf. Machine Learning, vol. 119, pp. 7076–7087, 2020.

[32] R. Verma and E. Nalisnick, "Calibrated Learning to Defer with One-vs-All Classifiers," in Proc. 39th Int.

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Conf. Machine Learning, vol. 162, pp. 22184–22202, 2022.

[33] National Institute of Standards and Technology, Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1, Jan. 2023.

[34] National Institute of Standards and Technology, Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile, NIST AI 600-1, July 2024.

[35] X. Sun, Y. Lu, and J. Chen, "Controllable Long-Term User Memory for Multi-Session Dialogue: Confidence-Gated Writing, Time-Aware Retrieval-Augmented Generation, and Update/Forgetting," JACS, vol. 3, no. 8, pp. 9-24, Aug. 2023, doi: 10.69987/JACS.2023.30802.

[36] M.-J. Kuo, D. Zheng, and J. Hires, "Federated Topic-Preference Learning for Knowledge-Grounded Chat with Differential Privacy," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 385-401, Aug. 2025, doi: 10.51903/jtie.v4i2.502.

[37] Y. Zhang and H. Zhang, "A Therapist-Facing Session Copilot for Live Counseling Support: Reasoning-Guided Retrieval and Ranking from Multi-Turn Counseling Dialogues," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 464-486, Aug. 2025, doi: 10.51903/jtie.v4i2.547.

[38] Q. Xin, "Behavior Retrieval plus Response Generation for Interpretable Conversational Personalized Recommendation," IJEEPSE, vol. 9, no. 2, pp. 120-136, Jul. 2026, doi: 10.31258/ijeepse.9.2.120-136.

[39] B. Zhang, H. Rao, and D. Zhao, "Evidence-Grounded RAG for Cloud-Native DevOps: Hallucination-Resistant AIOps Question Answering over Private Operations Documents," JACS, vol. 4, no. 3, pp. 109-125, Mar. 2024, doi: 10.69987/JACS.2024.40308.

[40] C. Li, J. Bai, and S. Wang, "Evidence-Chain Reliable RAG: Word-Level Hallucination Detection, Source Attribution, and Provenance Explanation for LLM Applications," JACS, vol. 4, no. 2, pp. 76-92, Feb. 2024, doi: 10.69987/JACS.2024.40207.

[41] G. Liu, C. Li, and E. Zhang, "OpsLLM for Cloud Incident Triage: Bilingual RAG-Based Root Cause Analysis and Alert Summarization for AI Infrastructure Operations," JACS, vol. 4, no. 4, pp. 97-111, Apr. 2024, doi: 10.69987/JACS.2024.40408.

[42] Z. S. Zhong, J. Chen, E. Zhong, and X. Sun, "Evidence-Calibrated RAG for Unanswerable Question Answering: Retrieval Coverage, Abstention Calibration, and Hallucination-Proxy Analysis on SQuAD 2.0," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 502-520, Aug. 2025, doi: 10.51903/jtie.v4i2.536.

[43] J. Jin, "Evidence-Chain Reliable RAG: Hallucination Detection, Source Attribution, and Deterministic Provenance Explanations," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 520-533, Aug. 2025, doi: 10.51903/jtie.v4i2.535.

[44] W. Su, S. Chen, and C. Zhao, "Budgeted Multi-Hop Retrieval Agent for Compositional Question Answering: A Retrieval-Policy Evaluation on the Official MultiHop-RAG Benchmark," J. Technol. Informatics Eng., vol. 4, no. 3, pp. 649-662, Dec. 2025, doi: 10.51903/jtie.v4i3.543.

[45] R. Zhang, Z. Wen, C. Wang, C. Tang, P. Xu, and Y. Jiang, "Quality Analysis and Evaluation Prediction of RAG Retrieval Based on Machine Learning Algorithms," arXiv:2511.19481, 2025, doi: 10.48550/arXiv.2511.19481.

[46] Y. Li, "Findable then Explainable: Retrieval-Summary Integration for Code Intelligence on a Lightweight CodeSearchNet Subset," JACS, vol. 4, no. 7, pp. 65-82, Jul. 2024, doi: 10.69987/JACS.2024.40706.

[47] G. Liu, S. He, and I. Liu, "LLM-Augmented Multi-Source Root Cause Attribution for CPU and Network Faults in Microservices," JACS, vol. 3, no. 6, pp. 39-57, Jun. 2023, doi: 10.69987/JACS.2023.30604.

[48] Q. Xin, "Explaining OpenStack Failure-Injection Log Anomalies with Retrieved Normal Prototypes," Emerg. Inf. Sci. Technol., vol. 6, no. 2, pp. 125-146, Nov. 2025, doi: 10.18196/eist.v6i2.31232.

[49] X. Sun, Z. S. Zhong, and Q. Wu, "Retrieval-Grounded HDFS Log Anomaly Detection and Deterministic Failure Narrative Generation," J. Comput. Syst. Appl., vol. 3, no. 1, pp. 15-30, Jun. 2026, doi: 10.64229/j6d7fr94.

[50] B. Zhang, X. Sun, G. Liu, and B. Zhou, "LLM-Style DevOps Copilot for Cloud-Native Troubleshooting: Retrieval-Augmented Runbook Generation and Command-Safety Evaluation," J. Technol. Informatics Eng., vol. 5, no. 2, pp. 104-118, Aug. 2026, doi: 10.51903/jtie.v5i2.534.

[51] D. Zheng, C. Li, and H. Davidson, "Continual Red-Teaming for In-the-Wild Jailbreaks via Online Guardrail Updates and Guardrail Distillation," JACS, vol. 3, no. 2, pp. 35-49, Feb. 2023, doi: 10.69987/JACS.2023.30203.

[52] W. Su, H. Rao, and E. Ma, "Privacy and Data-Integrity Risk Cards for LLM Agents: A UI/UX Design Framework for Secure Human Oversight under Prompt-Injection Attacks," Int. J. Graph. Des., vol. 4, no. 1, pp. 186-191, Apr. 2026, doi: 10.51903/ijgd.v4i1.3699.

[53] D. Zheng, B. Zhang, and J. Geibel, "VerifySafe: Toxicity-Safe Agent Responses under Adversarial Prompts with Evidence-Based Self-Verification," JACS, vol. 4, no. 1, pp. 67-82, Jan. 2024, doi: 10.69987/JACS.2024.40106.

[54] Z. S. Zhong, C. Li, and H. Rao, "Trajectory Reliability Prediction for Generalist AI Agents: Tool-Use Failure Analysis and Success Forecasting on ZClawBench," J. Technol. Informatics Eng., vol. 5, no. 1, pp. 341-360, Apr. 2026, doi: 10.51903/jtie.v5i1.539.

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

[55] D. Zheng and C. Li, "Behavior-Level Jailbreak Resistance via Multi-Stage Refusal + Utility Preservation," JACS, vol. 4, no. 1, pp. 83-99, Jan. 2024, doi: 10.69987/JACS.2024.40107.

[56] S. He, H. Tu, and I. Liu, "Safe PD Capacity Forecasting with Time-Series Foundation Models and Calibrated Uncertainty for Heterogeneous GPU Clusters," JACS, vol. 3, no. 4, pp. 48-66, Apr. 2023, doi: 10.69987/JACS.2023.30404.

[57] S. Chen, S. He, and E. Sun, "Risk-Bounded GPU Resource Oversubscription via Conformal Demand Envelopes in Production AI Clusters," JACS, vol. 4, no. 5, pp. 119-134, May 2024, doi: 10.69987/JACS.2024.40509.

[58] S. He, X. Chang, and E. Sun, "Cross-Cloud Transfer Learning for AI Training Capacity Forecasting under Workload and Topology Distribution Shift," JACS, vol. 4, no. 1, pp. 100-120, Jan. 2024, doi: 10.69987/JACS.2024.40108.

[59] S. Zhao, J. Bai, and D. Roberson, "Multi-Horizon GPU Demand Forecasting with Workload Semantics and Operational Risk Curves: An Empirical Study on Alibaba Clusterdata GPU Trace," J. Technol. Informatics Eng., vol. 4, no. 3, pp. 544-571, Dec. 2025, doi: 10.51903/jtie.v4i3.498.

[60] S. He, C. Li, and H. Rao, "Few-Shot Cold-Start Workload Forecasting for New AI Inference Tenants with Time-Series Foundation Models," J. Technol. Informatics Eng., vol. 4, no. 1, pp. 306-324, Apr. 2025, doi: 10.51903/jtie.v4i1.546.

[61] J. Nie, Y. Shi, and L. Zhao, "Token-Burst-Aware Capacity Planning for LLM Inference Services: Request Arrival, Token Demand, and Failure Risk Modeling from BurstGPT Traces," J. Technol. Inform. Eng., vol. 5, no. 2, pp. 142-164, Aug. 2026, doi: 10.51903/jtie.v5i2.565.

[62] S. Zhao, Y. Ren, and X. Chang, "Profit-Aware Spot GPU Admission Control with Cost-Sensitive Loss and Evidence-Grounded Policy Memos for AI Workload Supply-Demand Matching," J. Technol. Informatics Eng., vol. 5, no. 2, pp. 45-59, Jun. 2026, doi: 10.51903/jtie.v5i2.545.

[63] S. He, J. Nie, and C. Li, "Power-Aware Inventory Planning for AI Infrastructure Using Job-Level Forecasting and LLM Workload Explanations," J. Technol. Informatics Eng., vol. 5, no. 1, pp. 341-359, Apr. 2026, doi: 10.51903/jtie.v5i1.548.

[64] Y. Chen, Y. Zhang, D. Chau, and M. Sherman, "Credit Card Default Risk Tiering with Probability Calibration and Uncertainty-Driven Rejection: A Reproducible Study on the UCI Credit Card Clients Dataset," JACS, vol. 3, no. 4, pp. 31-47, Apr. 2023, doi: 10.69987/JACS.2023.30403.

[65] J. Jin, T. Huang, and S. Lu, "A Model-Risk-Friendly Probability of Default Workflow: Calibration, Distribution-Free Uncertainty Quantification, and SHAP Explanations on the UCI Credit Card Default Dataset," JACS, vol. 4, no. 6, pp. 74-85, Jun. 2024, doi: 10.69987/JACS.2024.40606.

[66] Q. Xin, "Explainable and Fair Credit Risk Scoring with Counterfactual Explanations: A Reproducible Evaluation on the German Credit Dataset (HELOC-Motivated)," J. Inf. Technol., vol. 14, no. 2, pp. 215-231, Jun. 2026, doi: 10.32664/j-intech.v14i02.2228.

[67] J. Jin, "Calibrated Resume-Job Matching for Trustworthy LLM-Assisted Recruiter Screening: Pairwise Matching, Probability Calibration, and Selective Refusal on Two Public Recruitment Datasets," J. Technol. Informatics Eng., vol. 4, no. 3, pp. 625-648, Dec. 2025, doi: 10.51903/jtie.v4i3.529.

[68] Q. Xin, "Uncertainty-Aware Late Fusion for 3D Perception (Confidence Calibration + Fusion Rule Learning)," J. Technol. Informatics Eng., vol. 4, no. 1, pp. 215-238, Apr. 2025, doi: 10.51903/jtie.v4i1.485.

[69] S. Lu and T. Zou, "Uncertainty-Aware Medical Vision-Language Classification on a Lightweight MedMNIST-Compatible Biomedical Patch Benchmark," J. Technol. Informatics Eng., vol. 5, no. 1, pp. 1-19, Jun. 2026, doi: 10.51903/jtie.v5i2.530.

[70] Q. Xin, "LiDAR-Camera Object-Level Fusion for Multi-Target Tracking Using JPDA and EKF: A Reproducible Empirical Study on a PandaSet-Parameterised Five-Sequence Dataset," J. Technol. Informatics Eng., vol. 5, no. 1, pp. 54-76, Apr. 2026, doi: 10.51903/jtie.v5i1.486.

[71] G. Mi, T. Ye, and D. Wood, "A Lightweight Medical Foundation Model for Cross-Modal Multi-Task Pretraining and Parameter-Efficient Few-Shot Transfer on MedMNIST," J. Technol. Informatics Eng., vol. 4, no. 3, pp. 572-589, Aug. 2025, doi: 10.51903/jtie.v4i3.492.

[72] J. Jin, "LLM-Style Evidence Cards for Scientific Search Interfaces: A UI/UX Design Framework for Retrieval Transparency, Ranking Trust, and Visual Evidence Hierarchy," Int. J. Graph. Des., vol. 3, no. 2, pp. 397-414, Oct. 2025, doi: 10.51903/ijgd.v3i2.3698.

[73] Z. S. Zhong, Q. Wu, and G. Mi, "Uncertainty-Aware Medical Image Explanation Cards: LLM-Generated Visual Explanations for AI-Assisted Radiology Interfaces," Int. J. Graph. Des., vol. 3, no. 2, pp. 415-436, Oct. 2025, doi: 10.51903/ijgd.v3i2.3616.

[74] B. Zhou, C. Li, and L. Liu, "Risk-Calibrated Patient-Facing AI Safety Cards: A UI/UX Design Framework for Rubric-Based Medical Risk Communication," Int. J. Graph. Des., vol. 3, no. 2, pp. 365-380, Oct. 2025, doi: 10.51903/ijgd.v3i2.3696.

[75] J. Nie, G. Liu, C. Li, and T. Zou, "Evidence-Constrained Incident Visualization Cards for Distributed Cloud Logs: A UI/UX Framework for Turning Hadoop, OpenStack, and ZooKeeper Logs into Actionable SRE Design Interfaces," Int. J.

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Graph. Des., vol. 4, no. 1, pp. 179-185, Apr. 2026, doi: 10.51903/ijgd.v4i1.3703.

[76] C. Li, G. Liu, and Z. Zhao, "Cost-Aware LLM-Style Routing for AIOps Log Analysis: Log Parsing, Anomaly Detection, Fault Diagnosis, and Incident Summarization on LogEval Task Files," J. Technol. Informatics Eng., vol. 5, no. 2, pp. 91-103, Jun. 2026, doi: 10.51903/jtie.v5i2.538.

[77] Q. Xin, "Hybrid Cloud Architecture for Efficient and Cost-Effective Large Language Model Deployment," J. Inf. Syst. Inform., vol. 7, no. 3, pp. 2182-2195, Sep. 2025, doi: 10.51519/journalisi.v7i3.1170.

[78] J. Nie and D. Zheng, "Noisy-Neighbor-Aware VM Degradation Risk Modeling with Unsupervised Residual Fusion," JACS, vol. 4, no. 4, pp. 112-123, Apr. 2024, doi: 10.69987/JACS.2024.40409.

[79] C. Wang, Z. Wen, R. Zhang, P. Xu, and Y. Jiang, "GPU Memory Requirement Prediction for Deep Learning Task Based on Bidirectional Gated Recurrent Unit Optimization Transformer," in Proc. 5th Int. Conf. Artificial Intelligence, Virtual Reality and Visualization (AIVRV), Chengdu, China, 2025, doi: 10.1109/AIVRV67401.2025.11350369.

[80] Q. Xin, "Self-Supervised Log Anomaly Detection with LogBERT-Style Transformers: Full Empirical Evaluation on a Reproducible SynHDFS Benchmark," J. Electr. Eng. Comput. Sci., vol. 11, no. 1, pp. 23-35, May 2026, doi: 10.54732/jeecs.v11i1.3.

[81] J. Bai, S. Chen, D. Zheng, and M.-J. Kuo, "Interpretable Attack-Chain Stage Detection from AWS CloudTrail Event Sequences via Linear Models and HMM Smoothing," Inf. Electr. Electron. Eng., vol. 6, no. 1, pp. 28-43, May 2026, doi: 10.33474/infotron.v6i1.24923.

[82] Q. Xin, "Log Anomaly Detection with Conformal Alert Control and Evidence-Grounded Incident Ticket Generation," AVITEC, vol. 8, no. 2, pp. 247-264, May 2026, doi: 10.28989/avitec.v8i2.3974.

[83] Z. Wen, R. Zhang, and C. Wang, "Optimization of Bi-Directional Gated Loop Cell Based on Multi-Head Attention Mechanism for SSD Health State Classification Model," in Proc. 6th Int. Conf. Electronic Communication and Artificial Intelligence (ICECAI), Chengdu, China, 2025, doi: 10.1109/ICECAI66283.2025.11171441.

[84] Q. Xin, "Host-Based Intrusion Detection with System Call Sequences: Window Localization and Forensic Narratives," AVITEC, vol. 8, no. 2, pp. 325-334, Jun. 2026, doi: 10.28989/avitec.v8i2.3973.

[85] R. Ma, L. Zhang, and A. Bai, "Cross-Dataset Parcel Workload Priors for Last-Mile Capacity Forecasting: Integrating Package Segmentation with Delivery Operation Signals," J. Technol. Inform. Eng., vol. 5, no. 1, pp. 441-473, Apr. 2026, doi: 10.51903/jtie.v5i1.564.

[86] Q. Xin, "Probabilistic Bike-Sharing Demand Forecasting under Changing Weather and Seasonal Regimes with Transformer-Based Models," Transport Findings, Mar. 2026, doi: 10.32866/001c.157499.

[87] J. Jin, T. Huang, and S. Lu, "Cost-Sensitive Learning, Simulated PU Learning, and One-Class Autoencoding for Extreme-Imbalance Credit Card Fraud Detection," JACS, vol. 4, no. 6, pp. 64-73, Jun. 2024, doi: 10.69987/JACS.2024.40605.

[88] S. Lu and D. Zhou, "TinyLLM-Assisted Intrusion Detection for Real-Time IoT Networks," JACS, vol. 4, no. 8, pp. 72-87, Aug. 2024, doi: 10.69987/JACS.2024.40809.

[89] Y. Chen, Y. Zhang, and M. Sherman, "Going Concern and Bankruptcy Prediction under Extreme Class Imbalance: Cost-Sensitive Learning, Resampling, and Focal Loss with Explainable Financial-Ratio Portraits," JACS, vol. 4, no. 4, pp. 80-96, Apr. 2024, doi: 10.69987/JACS.2024.40407.

[90] Y. Li and S. Lu, "Language-Guided Feature Selection for DDoS and Intrusion Detection on CICIDS2017," J. Technol. Informatics Eng., vol. 4, no. 1, pp. 284-305, Apr. 2025, doi: 10.51903/jtie.v4i1.531.

[91] Q. Xin, Z. Xu, L. Guo, F. Zhao, and B. Wu, "IoT Traffic Classification and Anomaly Detection Method Based on Deep Autoencoders," in Proc. 6th Int. Conf. Computing and Data Science (CDS), 2024.

[92] B. Wang, Y. He, Z. Shui, Q. Xin, and H. Lei, "Predictive Optimization of DDoS Attack Mitigation in Distributed Systems Using Machine Learning," in Proc. 6th Int. Conf. Computing and Data Science (CDS), pp. 89-94, 2024.

[93] J. Bai, H. Wang, Q. Wu, and B. Zhang, "Privacy-Robust Incrementality Estimation in Cookieless Settings via Uplift Modeling: Reproducible Evidence from the Hillstrom E-Mail Experiment," J. Technol. Informatics Eng., vol. 5, no. 1, pp. 17-38, Apr. 2026, doi: 10.51903/jtie.v5i1.468.

[94] H. Zhang, "DriftGuard: Multi-Signal Drift Early Warning and Safe Re-Training/Rollback for CTR/CVR Models," J. Adv. Comput. Syst., vol. 3, no. 7, pp. 24-40, 2023, doi: 10.69987/JACS.2023.30703.

[95] J. Mu, Y. Lu, and M. Smith, "LLM-Assisted Incrementality (Uplift) Modeling for Email Advertising: From Feature Interactions to Interpretable Audience-Creative-Channel Policies," JACS, vol. 3, no. 1, pp. 31-48, Jan. 2023, doi: 10.69987/JACS.2023.30103.

[96] T. Ye, J. Mu, and J. Hunter, "Off-Policy Evaluation and Conservative Policy Selection for Slot-Level Dynamic Bidding and Ranking on the Open Bandit Dataset (Small)," J. Technol. Informatics Eng., vol. 5, no. 1, pp. 178-199, Apr. 2026, doi: 10.51903/jtie.v5i1.503.

[97] H. Zhang, "Risk-Aware Budget-Constrained Auto-Bidding under First-Price RTB: A Distributional Constrained Deep Reinforcement Learning

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Framework," J. Adv. Comput. Syst., vol. 4, no. 6, pp. 30-47, 2024, doi: 10.69987/JACS.2024.40603.
[98] J. Bai and Q. Wu, "Privacy-Safe Marketing Mix Modeling and Budget Optimization under Identifier Loss: A Controlled Simulation Study," Int. J. Electron. Commun. Syst., vol. 6, no. 1, Jun. 2026, doi: 10.24042/ijecs.v6i1.30533.
[99] H. Zhang, "Counterfactual Learning-to-Rank for Ads: Off-Policy Evaluation on the Open Bandit Dataset," J. Adv. Comput. Syst., vol. 5, no. 12, pp. 1-11, 2025.
[100] Y. Chen, S. Zhou, and E. Lin, "Accounting-Aware Evidence Retrieval for Institutional Due Diligence of Tokenized Trade Receivable RWA," J. Technol. Informatics Eng., vol. 4, no. 3, pp. 649-663, Dec. 2025, doi: 10.51903/jtie.v4i3.542.
[101] J. Li and A. Zhou, "Multi-Regulation RAG for AI Product Counsel: A Legal Governance Framework for Cross-Border Digital Commerces," Rule Law Stud. J., vol. 2, no. 2, pp. 105-123, Jun. 2026, doi: 10.64780/rolsj.v2i2.225.
[102] S. Zhou, Y. Chen, and K. Lee, "Accounting-Aware Evidence-Constrained Agents for Disclosure, Settlement, and Secondary-Market Risk Monitoring in Tokenized Assets," J. Technol. Informatics Eng., vol. 5, no. 2, pp. 60-74, Jun. 2026, doi: 10.51903/jtie.v5i2.544.
[103] G. Zhao, D. Zhang, and S. Meng, "LLM-Inspired Ontology-Based Semantic Enrichment for FinTech M&A Intelligence in SEC Structured Disclosures," J. Technol. Inform. Eng., vol. 5, no. 2, pp. 119-141, Aug. 2026, doi: 10.51903/jtie.v5i2.566.
[104] S. Meng, J. Chen, and I. Zheng, "LLM-Inspired Offline Reranking for Financial Search: Query Rewriting, Hybrid Retrieval, and Listwise Relevance Ranking on FiQA," J. Technol. Informatics Eng., vol. 5, no. 1, pp. 361-378, Apr. 2026, doi: 10.51903/jtie.v5i1.537.
[105] Z. Li, K. Zhang, and A. Wong, "Numerical-Reasoning Guardrails for a Quant Research Assistant: A Compact Reproducible Benchmark Using SEC and FRED Data," J. Technol. Informatics Eng., vol. 5, no. 2, pp. 75-90, Jun. 2026, doi: 10.51903/jtie.v5i2.541.
[106] K. Zhang, Y. Chen, and A. Qian, "Evidence-Grounded Accounting Disclosure Review Cards: A Visual Communication Framework for LLM-Style Explanations over SEC Financial Statements and Notes," Int. J. Graph. Des., vol. 3, no. 2, p. 395, Oct. 2025, doi: 10.51903/ijgd.v3i2.3710.
[107] Z. Li, S. Zhou, and Z. Zhou, "Financial Risk Dashboard Design for Institutional RWA Investors: Visual Hierarchy, Chart Comprehension, and Explainability in FinChart-Bench," Int. J. Graph. Des., vol. 3, no. 1, pp. 196-210, May 2025, doi: 10.51903/ijgd.v3i1.3715.
[108] B. Zhang, Y. Ren, and J. Zou, "LLM-Style Explainable E-Commerce Recommendation Cards: A UI/UX Design Framework for Trust-Calibrated Product Recommendation," Int. J. Graph. Des., vol. 3, no. 2, pp. 381-396, Oct. 2025, doi: 10.51903/ijgd.v3i2.3697.
[109] Q. Xin, "Self-Supervised Customer Representation Learning for Segmentation and Next-Purchase Prediction on UCI Online Retail," J. Inf. Technol., vol. 14, no. 1, pp. 20-37, Apr. 2026, doi: 10.32664/j-intech.v14i01.2229.
[110] X. Chang, Y. Lu, and Z. S. Zhong, "Review-Grounded Explainable Recommendation with Faithfulness Evaluation on Amazon Reviews," J. Electr. Eng. Comput. Sci., vol. 11, no. 1, pp. 9-22, May 2026, doi: 10.54732/jeecs.v11i1.2.
[111] K. Xu, H. Zhou, H. Zheng, M. Zhu, and Q. Xin, "Intelligent Classification and Personalized Recommendation of E-Commerce Products Based on Machine Learning," in Proc. 6th Int. Conf. Computing and Data Science (ICCDS), 2024.
[112] Q. Xin, "Auditable Automated Essay Scoring and Formative Feedback: A Rubric-Grounded Pipeline for Secondary and Higher Education," J. Appl. Artif. Intell. Educ., vol. 2, no. 1, Jul. 2026, doi: 10.66053/jaaie.v2i1.348.
[113] Y. Zhang and Z. Zhou, "Strategy-Aware Therapist Imitation for Emotional Support Dialogues: A Reproducible ESConv Study for LLM Response Control," Adv. Educ. Technol. Psychol., vol. 10, no. 2, pp. 92-97, 2026, doi: 10.23977/aetp.2026.100213.
[114] Q. Xin, "Early-Warning Analytics with LLM Intervention Rationales for Student Retention Decisions: Classroom Interaction Modeling with xAPI-Edu-Data and Dropout/Success Prediction," Interdiscip. J. Pedagog. Res. Media Technol., vol. 2, no. 1, Jun. 2026, doi: 10.64268/inspire.v2i1.117.
[115] J. Zhang, "Early Warning, Grade Prediction, and Teacher-Facing LLM-Ready Explanations toward an Open Volleyball Course: Reproducible Evidence from Four Public Education Datasets," J. Technol. Informatics Eng., vol. 5, no. 2, pp. 20-44, Jun. 2026, doi: 10.51903/jtie.v5i2.525.
[116] Y. Zhang and H. Zhang, "Visualizing the Right Counseling Support: Evidence-Linked Recommendation Cards for Explainable Mental Health Intake Interfaces," Int. J. Graph. Des., vol. 3, no. 1, pp. 214-229, May 2025, doi: 10.51903/ijgd.v3i1.3722.
[117] Y. Chen and H. Xu, "Trust-Calibrated Multilingual RAG for Humanitarian Information Platforms: Empirical Evaluation on OMoS-QA for Migration Information Access," Int. J. Graph. Des., vol. 4, no. 1, pp. 141-164, Apr. 2026, doi: 10.51903/ijgd.v4i1.3552.
[118] Q. Wu, S. Meng, and J. Zhao, "Text-Grounded LLM-Assisted Design Rationale Interfaces: Turning Advertising Layout Metadata into Explainable UI/UX Decision Cards," Int. J. Graph. Des., vol. 3, no. 1, pp. 216-240, May 2025, doi: 10.51903/ijgd.v3i1.3713.
[119] H. Xu, Y. Chen, and A. Med, "Automatic Detection and Explanation of Dark Patterns from Interface Microcopy: Empirical Comparison of BERT-Style

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

Encoders, RoBERTa-Style Encoders, and LLM-Style Decoders on the ec-darkpattern Dataset," J. Technol. Informatics Eng., vol. 4, no. 3, pp. 590-612, Dec. 2025, doi: 10.51903/jtie.v4i3.491.

[120] H. Tu, S. Zhao, and A. Zhou, "Visual Brief Cards for Advertising Design: A Structured UI/UX Framework for Turning Creative Intentions into Graphic Design Decisions," Int. J. Graph. Des., vol. 3, no. 1, pp. 210-226, May 2025, doi: 10.51903/ijgd.v3i1.3714.

[121] Y. Chen and M. Li, "From Hand-Drawn Sketches to Interactive Web Prototypes: A Reproducible Vision-Language Approach with Structural and Visual Consistency Evaluation," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 364-384, Aug. 2025, doi: 10.51903/jtie.v4i2.490.

[122] Y. Li, S. Lu, and L. Zhao, "LLM-as-Design-Critic: Aligning AI-Generated UI Feedback with Human Graphic Design Judgment," Int. J. Graph. Des., vol. 3, no. 1, pp. 196-215, May 2025, doi: 10.51903/ijgd.v3i1.3661.

[123] H. Wang, Y. Ren, and X. Chang, "Layout-Aware Progressive PDF Rendering: AI Prioritization of PDF Slices to Reduce Time-to-Functional-First-Frame on FUNSD," J. Technol. Informatics Eng., vol. 4, no. 2, pp. 425-446, Aug. 2025, doi: 10.51903/jtie.v4i2.523.

[124] J. Mu, Y. Lu, and E. Hwang, "Structured Visual Brief Interfaces for Advertising Design: A UI/UX Framework for Turning Creative Intentions into Designer-Editable Graphic Design Cards," Int. J. Graph. Des., vol. 4, no. 1, pp. 192-208, Apr. 2026, doi: 10.51903/ijgd.v4i1.3702.

[125] Z. Ling, Q. Xin, Y. Lin, G. Su, and Z. Shui, "Optimization of Autonomous Driving Image Detection Based on RFAConv and Triplet Attention," in Proc. 2nd Int. Conf. Software Engineering and Machine Learning (SEML), 2024.

[126] Z. Zhong, M. Zheng, H. Mai, J. Zhao, and X. Liu, "Cancer Image Classification Based on DenseNet Model," J. Phys.: Conf. Ser., vol. 1651, no. 1, Art. 012143, Nov. 2020, doi: 10.1088/1742-6596/1651/1/012143.

[127] Q. Wu, G. Mi, and D. Wood, "Calibration-Light Subject-Independent Motor Imagery BCI via Self-Supervised Pretraining and Conformer," J. Technol. Informatics Eng., vol. 4, no. 1, pp. 239-262, Apr. 2025, doi: 10.51903/jtie.v5i1.493.

[128] J. Chen, J. Xiong, Y. Wang, Q. Xin, and H. Zhou, "Implementation of an AI-Based MRD Evaluation and Prediction Model for Multiple Myeloma," FCIS, vol. 6, no. 3, pp. 127-131, Jan. 2024, doi: 10.54097/zJ4MnbWW.

Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)