AGENT CARCINIZATION: WHEN METRICS CHANGE THE VERDICT ON LLM RUNTIME STRUCTURE

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent carcinization is the hypothesis that shared operational pressures favor recurring model-external control structures. Assessing their adaptive value requires distinguishing task benefits from rewards for performing the structures' own procedures. We audit 200 replay-verified episodes from four fixed models, four synthetic tasks and five runtime conditions. On the same three business tasks, the capability-gated minus schema-placebo contrast changes from +27.08 percentage points under historical success to -29.86 under a separately implemented outcome ledger. Requiring completion with explicit finish or no recorded violation attenuates the contrast to -13.19 or -2.08 points. The unconstrained reversal survives every single task or model deletion; the violation-constrained contrast changes sign under model deletion. The ledger omits explicitly requested procedures and is not a replacement ground truth. An external retail comparison adds 160 Youtu episodes and a separate 32-episode Qwen check on eight tasks. Completion contrasts are -2.50 and 0.00 points, but only 4/80 Youtu pairs and no Qwen pairs encounter the boundary. These results do not replicate the reversal or establish equivalence. Scripted probes and supplementary cancellation episodes distinguish immediate blocking from post-denial recovery. Our contribution is a trace-grounded audit of apparent runtime benefits through endpoint provenance, matched rescoring, constraint sensitivity and intervention exposure. Independent human endpoint adjudication remains future work.

1 INTRODUCTION

Long-running agents face recurring engineering demands: preserving state, recovering from failures, controlling tool authority, and checking outcomes. We call the hypothesis that such pressures favor recurring model-external control structures *agent carcinization*. The term borrows the distinction between repeated form and shared ancestry from biological carcinization [1]; it does not imply a literal evolutionary process or inevitable growth in software complexity. The candidate shared structure is a set of runtime responsibilities, not one implementation or a requirement for multi-agent orchestration.

The empirical question here is whether the apparent benefit of a more structured runtime survives changes in how benefit is measured. A harness may be rewarded for inspecting, checkpointing, or validating simply because its evaluator counts those procedures. Yet those procedures can also be legitimate user requirements. Distinguishing these cases matters before treating a score improvement as evidence of useful architectural adaptation.

We examine this problem through a replay-verified case study of 200 episodes from four fixed local models, four synthetic tasks, and five runtime conditions. The motivating comparison is capability gating against a schema placebo. We ask whether the direction of a reported difference persists across explicit endpoint contracts and whether the result depends on a single task or model. Our analysis is post hoc and descriptive.

The contribution is a linked audit, rather than the generic observation that metrics matter: trace-level counterexamples are paired with their observed incidence; fixed trajectories are compared under outcome and constrained contracts; deletion checks test dependence on observed strata; and

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.

exposure audits establish whether an external comparison actually exercises the mechanism. The local reversal, its attenuation under constraints, and its absence externally are all retained as results.

Our central claim is bounded: endpoint and comparator choices change conclusions in this corpus. Neither a favorable procedural score nor a favorable business-completion score alone establishes overall runtime utility. Changing the measured construct also does not show that one score is universally correct.

What would make the structure adaptive? We distinguish three propositions: recurring runtime form, its emergence under common pressures rather than inheritance or diffusion, and a useful functional effect. Each needs different evidence. For the third, we propose reporting a benefit vector of completion, required compliance, recovery and realized boundary effects alongside cost, with acceptable trade-offs fixed before evaluation. More components or more procedural events alone do not establish improvement. Redundant controls may be removed as tasks or models change; such simplification is compatible with the conditional hypothesis. Appendix A.1 specifies discriminating tests.

2 Related Work

Tool use, reasoning-action loops, reflection, and memory motivate increasingly structured agent runtimes [2, 3, 4, 5, 10, 11]. AgentBench, WebArena, OSWorld, and SWE-bench assess agents in interactive environments [6, 7, 8, 9]. These benchmarks address broader capabilities than the small tool worlds studied here. We examine the measurement contract of a runtime comparison rather than propose a replacement benchmark.

Source-based architectural studies and harness surveys characterize recurring runtime responsibilities [12, 13, 14]. Work on automatically changing harnesses highlights the need for matched feedback budgets and held-out evaluation [15]. Our analysis addresses a complementary issue: even with fixed trajectories and budgets, changing the endpoint can change the direction of the reported difference. Zhu et al. identify task-setup and reward-design problems in agentic benchmarks and propose the Agentic Benchmark Checklist [16]. This is closely related prior work: evaluation validity and reward defects are not new claims here. Our narrower contribution is a matched-trajectory analysis of comparator-dependent sign reversals, their observed incidence, and exhaustive single-stratum deletion checks. The database-state evaluator in τ-bench [17] also provides an external contract against which to test whether our local inspection dependence transfers.

3 Measurement Contracts

We distinguish five quantities. *Business completion* records the intended state change. *Instruction compliance* records required procedures as well as outcomes. *Closure* records explicit termination. *Attempted violations* record disallowed requests before enforcement; *realized effects* include resulting state changes or successful protected information reads. A read need not change the database to cross an information boundary. Runtime costs are reported alongside these quantities.

A process-inclusive score can be appropriate when the procedures themselves are required. The risk arises when it is interpreted as business completion, safety, or audit quality without checking its contract. For example, counting repeated inspection measures activity rather than coverage; counting only handler-level violations can omit attempts rejected by an earlier gate.

We use three checks: an executable contract probe establishes whether a behavior is possible, trajectory incidence establishes whether it occurred in the corpus, and paired rescoring establishes how a chosen endpoint changes a comparison. A prospective run is required to study behavior under a changed environment. None of these checks substitutes for the others.

All reported labels are computed from recorded events or evaluator outputs by the study authors. Separate ledger implementation and tool-state reconstruction check computation, not independent endpoint validity. Independent human adjudication is future work; no human review results enter this paper. Appendix A.3 specifies label sources, denominators, and dependencies. Mechanism probes establish behavior at supplied states, not overall performance or population safety.

4 BEHAVIORAL STUDY

4.1 TASKS, CONDITIONS, AND EXECUTION

A deterministic tool world receives JSON actions from an LLM. Four tasks cover a payment retry with idempotency, a tenant-scoped record update, webhook processing with replay handling, and context continuity after compression. The model chooses actions and arguments; the runtime exposes state, enforces the selected controls, and returns tool events. Outcomes are computed from the trajectory and terminal state.

Five conditions compare distinct control surfaces: bare, bare with an explicit closure-action instruction, state only, capability-gated, and schema placebo. The gated/placebo pair shares the intended action-schema structure and state fields, while enabled actions and enforcement differ. It is a schema control, not a guarantee of equal information: observation filtering and retained history can also differ.

The corpus contains ten blocks, each with four tasks and five conditions, for 200 episodes. Youtu 4B contributes four blocks, Qwen3.5-9B two, Qwen3 4B three, and Youtu 35B one. Every block uses temperature 0.2, a 14-turn limit, and a 1,200-output-token cap per model request. Appendix A records model identifiers and block composition. Schedule seeds label blocks; they are not verified independent model-sampling seeds.

4.2 ENDPOINTS AND AGGREGATION

We distinguish the endpoints before interpreting their scores:

- **Historical success** requires successful inspection, required task actions, valid validation, no rollback, and no counted violation. It mixes process compliance with task completion.
- **Closure** records explicit finish. **Legacy audit** counts successful inspection, checkpoint, and validation actions, divided by three and capped at one; repeated actions count.
- **Recovery** is evaluated on retry and webhook. **Legacy safety** is a graded violation score; **no violation** is a separate binary indicator. Both safety summaries use retry, tenant, and webhook.
- **Cost** records actions, tokens, and invalid responses. A post-hoc **terminal-record** proxy requires inspection and valid validation immediately before finish.

Replay checks all recorded actions, events, terminal states, scores, and observations. No duplicate trial identifiers, missing task–condition cells, or replay mismatches occur. Scripted probes test endpoint behavior and are excluded from the 200 model episodes.

For each endpoint, condition differences are computed on the same eligible tasks, averaged within blocks, then within models, then equally across the four models. Incidence rates instead use raw counts with explicit denominators. These are descriptive fixed-corpus estimates; neither models nor tasks are random population samples, and one model has only one block.

4.3 OUTCOME-LEDGER SENSITIVITY

A separate event ledger reconstructs business completion for retry, tenant, and webhook without calling the historical scorer or validation function. It requires a single settled charge, an update authorized by a previously issued nonempty credential, or one webhook effect with acknowledged replay, respectively. It tracks rollback and failed-action side effects. Context is excluded because existing outputs cannot establish value-level recall.

The ledger intentionally omits inspection, validation, finish, and rejected attempts from its completion criterion. It therefore measures a narrower outcome construct, not full instruction compliance or externally validated safety. We also remove only the inspection requirement from historical success, leaving every other condition unchanged. This single-factor sensitivity distinguishes one procedural dependency from the broader change in construct.

All three business-task requests explicitly require validation and finish. Thus omitting these is a deliberate estimand change, not correction of an irrelevant requirement. Inspection is present in

scorer-required actions but not named in these goal strings; this alone cannot settle its necessity under the full prompt and environment. Appendix A.2 traces these distinctions to the task definitions and records unresolved judgments.

4.4 External prospective design

The separate external protocol uses the original retail tasks, tools, database, policy and state evaluator from the pinned τ-bench source. It selects two distinct user/order clusters from each of four operation families: cancellation, return, exchange and pending-item modification. Ten repeated schedules per task and condition give 160 planned episodes. Both arms use identical prompts and tool schemas; one enforces simulator identity binding and resource ownership before protected calls, while the other records the same attempts without blocking them. Original tool preconditions remain active in both arms. The binding is an environment identity resolution, not real-world authentication.

The agent and simulated user share a local Youtu 4B model with a recorded weight digest. The original agent and simulated-user loops execute through a native local transport, with 30 agent turns and full tool observations. The user simulator receives the task instruction, which is withheld from the agent along with reference actions. The four external endpoints are original state completion, completion with dialogue termination, completion without a recorded boundary attempt, and completion without a recorded boundary effect. Primary attempt counts refer to the first tool call selected by the upstream loop, before enforcement; a later sensitivity audit classifies discarded proposals at the pre-turn identity state. These narrow boundary measures do not cover all retail policies, including the semantics of user confirmation. This is a modified-harness external-task experiment, not an official benchmark score. Appendix A.13 records development runs, budgets, and excluded protocols.

A subsequent configuration-sensitivity protocol fixes Qwen3.5-9B for both agent and simulated user on the same eight tasks, with two paired schedules per task and arm (32 planned episodes). It retains the execution implementation, prompts, tools, endpoints and budgets. The matched Youtu comparison uses the same task, condition and seed identifiers from repetitions 0 and 1 of the original corpus. Changing both roles tests the joint configuration, not the isolated agent model. This limited extension is specified after the original external results and is reported separately.

5 Results

5.1 Matched endpoint sensitivity

Table 1 presents four co-reported endpoints on exactly the same three business tasks and 30 episodes per condition. Historical success favors gating, whereas business completion favors the placebo. However, requiring both completion and no historical recorded violation reduces the difference from -29.86 to -2.08 percentage points. This qualification is central to the interpretation: the unconstrained completion contrast does not establish superiority under task constraints. Historical violation counts are rejection-layer dependent, so the constrained endpoint does not establish equivalent safety either.

Table 1: Core historical endpoints on matched trajectories. Rates use equal model weights; differences are gated minus placebo in percentage points. No endpoint is designated universal ground truth.

Endpoint	Gated (%)	Placebo (%)	Difference
Historical success	27.08	0.00	+27.08
Business completion	54.17	84.03	-29.86
Completion and explicit finish	54.17	67.36	-13.19
Completion and no recorded violation	54.17	56.25	-2.08

Table 2 retains all four comparators and the inspection-only sensitivity. The ledger direction also changes against closure-action and state-only controls, but not against bare.

Removing only inspection increases the gated advantage against every comparator. Consequently, inspection dependence alone cannot explain the ledger reversal. The broader endpoint change also

Table 2: Endpoint sensitivity on the matched three-task subset. Entries are capability-gated minus control in percentage points, with equal model weights. Each column uses the same trajectories.

Control	Historical success	Without inspection	Ledger completion
Bare	+27.08	+47.92	+0.69
Closure action	+27.08	+38.89	-15.97
State only	+27.08	+41.67	-5.56
Schema placebo	+27.08	+39.58	-29.86

removes validation, other procedural requirements, and violation-based failure. These conditions overlap and are not additive causal contributions.

These differences motivate separate review of which procedures are necessary under the original request. Rescoring fixed trajectories does not establish how models would behave under a repaired environment.

5.2 Comparator sensitivity

Figure 1 reports the same 200-episode corpus throughout. Relative to bare, capability gating increases historical success by 0.203 and closure by 0.333, with recovery +0.052 and legacy safety -0.010. On safety-eligible tasks, the zero-violation fraction decreases by 0.132. Mean token use rises by 1,921 relative to bare but falls by 259 relative to the schema placebo. Thus the cost conclusion changes with the baseline.

The apparent legacy-audit gain is +0.344 against bare and +0.089 against placebo. The terminal-record proxy yields +0.203 and +0.141, respectively. Because both measures count procedural events, neither establishes general audit quality. These results describe execution differences; they do not isolate a causal effect of enforcement.

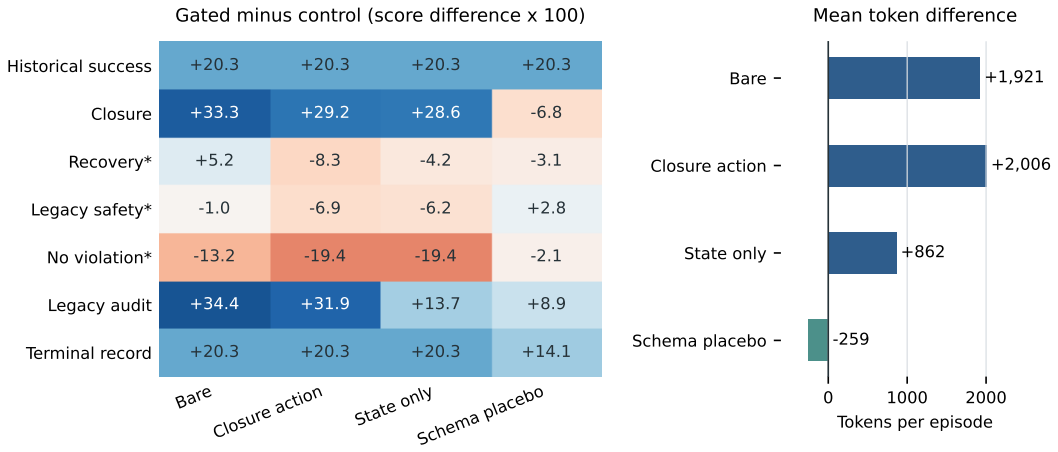


Figure 1: Capability-gated minus control on historical endpoints. The left panel shows score differences multiplied by 100; the right shows tokens per episode. Tasks and blocks are averaged within each model before equal model weighting. Recovery uses two tasks; safety and no violation use three; other rows use four. Positive values mean more of the named quantity, not necessarily improvement.

5.3 Procedural scores and observed outcomes

Repeated inspection can maximize legacy audit without completing any task. This is an executable counterexample; the corpus separately contains 48/200 episodes with maximal legacy audit and

historical failure. Removing only the inspection requirement changes 51/200 success labels. These observations establish that procedural requirements materially affect this corpus, rather than merely presenting a hypothetical scorer weakness.

Other counterexamples have different incidence. No unissued null-credential attempt is observed in the 50 tenant episodes, and no historically successful episode lacks finish. Unfiltered inspection history reaches a later model request in 28/80 gated-or-placebo episodes. This last finding limits interpretation of the placebo as an information-matched control.

5.4 Task and model deletion checks

We remove each business task in turn and recompute all four control comparisons, preserving equal model weights. Against placebo, historical-success differences remain positive (+3.12 to +40.62 points), while ledger differences remain negative (-44.79 to -19.79). Removing each model in turn likewise preserves this reversal: historical differences range from +22.22 to +36.11 points and ledger differences from -37.04 to -25.00. These are descriptive sensitivity ranges, not confidence intervals.

The same conclusion does not hold uniformly across comparators. Ledger differences against bare range from -11.46 to +12.50 across task deletions; against state only they range from -20.83 to +12.50. Model deletion also changes the sign for these two controls. Thus the placebo reversal is not driven solely by one observed task or model, but the corpus does not support a general claim that gating harms outcomes.

The two constrained core endpoints are less stable than unconstrained completion. Against placebo, completion with finish ranges from -19.79 to 0.00 points under task deletion and -17.59 to -6.48 under model deletion. Completion with no recorded violation ranges from -3.12 to 0.00 under task deletion and -11.11 to +8.33 under model deletion. In particular, the near-zero full-corpus contrast cannot support a stable ordering of the conditions under the recorded-violation constraint.

5.5 External evaluator contract checks

We exercise the unmodified τ-bench retail evaluator and tools at a fixed source revision on the first five test tasks satisfying an explicit mutation-only evaluation criterion (Appendix A.11). Across 20 scripted checks, reference actions succeed, three repeated read-only inspections alone fail, adding those inspections preserves success, and removing read-only actions from the reference sequence preserves success. All 20 checks match these expectations.

This is negative transfer evidence for the specific local inspection dependence: these external state-based checks do not require inspection or reward its repetition. The reference actions already contain the necessary arguments, so omitting reads here says nothing about an agent's ability to discover those arguments. These are offline contract checks, not model-performance measurements or a comprehensive audit of τ-bench.

5.6 External repeated comparison

All 160 planned episodes completed: eight retail tasks, ten paired schedules per task, and two conditions. All 4,336 model calls and the resulting tool events, state hashes and endpoints passed cached-response replay; direct tool reconstruction also matched. Table 3 reports the four endpoints. Gating yields 27/80 state completions versus 29/80 for control, and 21/80 completions with dialogue termination versus 23/80. Requiring no recorded boundary attempt or effect gives 27/80 in both arms. These equal observed constrained rates do not establish equivalence.

Only four of 80 paired schedules differ on state completion: three favor control and one favors gating. The paired-difference standard deviation is 22.36 percentage points for completion and completion with termination, and 15.91 for the two boundary-constrained endpoints. Across task/arm strata, ten schedules produce six to ten distinct tool-action sequences after excluding dialogue wording. Task-level rates vary sharply (Appendix Table 10); repeated schedules do not constitute new task samples.

Table 3: External prospective comparison: eight fixed retail tasks, ten repetitions per condition and task (80 episodes per condition). Equal task weights. These endpoints use the external state evaluator and a narrow identity boundary; they are not full policy compliance.

Endpoint	Gated (%)	Control (%)	Difference (pp)
Original state completion	33.75	36.25	-2.50
Completion and dialogue termination	26.25	28.75	-2.50
Completion, no recorded boundary attempt	33.75	33.75	+0.00
Completion, no recorded boundary effect	33.75	33.75	+0.00

Table 4: Matched configuration sensitivity: the same eight tasks and two paired schedules (16 episodes per arm per configuration). G/C are gated/control rates in percent; Δ is gated minus control in percentage points. Both agent and simulated user change model. Youtu entries are a fixed 32-episode subset of the original 160, not additional trials.

	Youtu 4B			Qwen3.5-9B		
Endpoint	G	C	Δ	G	C	Δ
Original state completion	31.25	31.25	+0.00	43.75	43.75	+0.00
Completion and dialogue termination	18.75	18.75	+0.00	43.75	43.75	+0.00
Completion, no recorded boundary attempt	31.25	31.25	+0.00	43.75	43.75	+0.00
Completion, no recorded boundary effect	31.25	31.25	+0.00	43.75	43.75	+0.00

5.7 Second configuration and intervention exposure

In the matched 32-episode configuration check, Qwen3.5-9B gives 7/16 completions in each arm, with zero paired differences across all four endpoints. The matched Youtu subset gives 5/16 completions in each arm and likewise zero paired differences; completion with termination is 3/16 in each arm. These are configuration-specific observations, not equivalence tests: two repeats per task cannot resolve small effects, and both conversational roles changed. The Qwen runs recorded no boundary attempts or effects and no budget exhaustion. The full ten-repeat Youtu corpus remains the primary external comparison.

A post-hoc paired exposure audit shows that the gate never blocks a call in the 16 Qwen pairs, all of which have identical action sequences and terminal states across arms. Thus their zero contrast does not test behavior after an enforced denial. In Youtu, only 4/80 pairs encounter the boundary and all four contain a blocked gated call. Two of the four endpoint-discordant pairs have no boundary exposure in either arm. Equal schedule seeds therefore do not remove observed run-to-run variation; the small aggregate contrast cannot be assigned entirely to denial events. Appendix A.16 reports action, dialogue and state agreement separately.

In the full Youtu corpus, boundary attempts occur in 4/80 gated and 3/80 control episodes; realized effects occur in 0/80 and 3/80. The four blocked calls and low event counts support only narrow implementation claims. Auditing all 232 discarded proposals finds no additional boundary violation. Budget exhaustion, unsuccessful termination and the interrupted-pair sensitivity are reported in Appendix A.13.

For the full Youtu corpus, task deletion gives completion contrasts from -4.29 to +1.43 points; family deletion gives -5.00 to +1.67. Completion with termination has the same ranges. Boundary-constrained contrasts range from -1.43 to +1.43 under task deletion and -1.67 to +1.67 under family deletion. The observed task mix therefore does not support a stable ordering.

5.8 Expanded external mechanism checks

A post-hoc intervention study includes all 39 retail test tasks satisfying the one-mutation, no-text-output, no-reference-dialogue eligibility rule, covering 30 user/order clusters and six action families. Each reference mutation is tested after its reference prefix with correct, absent or incorrect simulator identity binding, with and without gating: 234 scripted probes. Gating blocks all 78 absent/wrong-identity probes and none of the 39 correct-identity probes. Without gating, 38/39 probes in each identity category change state. These oracle-argument probes establish implementation behavior

under supplied identity states, not model performance, real-world authentication robustness or independent replication.

The remaining task, index 106, exposes an external evaluator counterexample. Its reference exchange fails for insufficient gift-card balance and leaves the database unchanged. The original evaluator then rewards an untouched initial environment with 1 because it matches that ineffective reference state. A follow-up census of the same 39 tasks finds this behavior only in task 106; all other untouched environments score 0. None of the eight behavioral tasks is affected. This separates one executable scoring failure from its incidence in a restricted task set; it is not an estimate over the entire benchmark. Appendix A.14 gives the protocol and intervention counts.

A later 24-episode cancellation development experiment is reported only as supplementary mechanism evidence (Appendix A.4). Its scripted setup and two clusters do not expand the primary performance sample or provide an independent replication.

6 Discussion and Limitations

The strongest result is a within-corpus reversal across explicitly different measurement contracts, substantially attenuated when no recorded violation is also required. The historical-success versus unconstrained-completion reversal remains under every single task or model deletion for the placebo comparison. This robustness does not make the ledger externally validated, turn a post-hoc analysis into a confirmatory test, or identify a causal effect of enforcement.

The historical gated/placebo comparison changes enabled actions and may expose different observations. Both are relevant runtime interventions, but their effects cannot be separated by rescoring. Our ledger is separately implemented yet still depends on the simulator's event truth. Its completion criterion excludes procedures and rejected attempts; a negative difference on this endpoint is not evidence of worse safety.

The four synthetic tasks and four fixed models are not population samples. Three tasks support the ledger analysis; value-level context recall is unidentifiable in retained records. The largest model has one block, and recorded historical model tags are mutable. Single-deletion analyses test dependence on observed strata, not unseen task or model generalization. The external comparison adds repeated behavior on eight selected tasks with Youtu and a smaller Qwen configuration check, each using the same model for agent and user, after earlier engineering exposure to test tasks. Boundary exposure is rare in Youtu and absent in Qwen. These small observed contrasts do not replicate the historical reversal or establish equivalence. Independent expert assessment of the endpoint contracts and broader replication remain necessary.

For agent carcinization, the distinction is between useful runtime structure and an evaluation contract that rewards the same structure. The observed contrast spans -29.86 to +27.08 points across four substantively different endpoints; this is a descriptive contract-sensitivity range, not uncertainty about one estimand. Its narrowing under a violation constraint shows why the largest reversal alone would overstate the case. The external non-reversal and low exposure limit transfer of the empirical pattern while motivating reuse of the audit procedure. No universal architecture ranking follows.

7 Conclusion

Agent carcinization provides a motivating question: when do recurring control structures constitute useful adaptation? A replay-verified analysis of 200 episodes shows why a scalar runtime score is insufficient to answer it. Against a schema placebo, historical success favors gating by 27.08 points while ledger completion disfavors it by 29.86; adding finish or no recorded violation reduces that disadvantage to 13.19 or 2.08 points. The defensible contribution is an audit linking requirement provenance, fixed-trajectory contrasts, constraint sensitivity and actual mechanism exposure. It supports evaluating runtime responsibilities by explicit task contracts and trade-offs, with independent semantic adjudication and transfer tests as the next evidential steps.

Reproducibility Statement

The local evidence bundle contains historical trajectories, source hashes, task definitions, replay analyses, 160 Youtu external trajectories and 32 separate Qwen trajectories. Historical rescoring, external replay and the paired exposure audit require no model calls. Excluded pilots and the interrupted prefix remain separately identified. Appendix A records model composition and endpoint definitions.

AI Use Statement

Generative AI tools assisted drafting, code inspection, experiment execution, and deterministic audits. The authors remain responsible for the code, references, empirical interpretations, and submission.

References

[1] J. M. Wolfe, J. Luque, and H. D. Bracken-Grissom. How to become a crab: Phenotypic constraints on a recurring body plan. *BioEssays*, 43(5):e2100020, 2021. doi:10.1002/bies.202100020.

[2] Z. Xi et al. The rise and potential of large language model based agents: A survey. arXiv:2309.07864, 2023.

[3] S. Yao et al. ReAct: Synergizing reasoning and acting in language models. In *ICLR*, 2023.

[4] T. Schick et al. Toolformer: Language models can teach themselves to use tools. arXiv:2302.04761, 2023.

[5] N. Shinn et al. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*, 2023.

[6] X. Liu et al. AgentBench: Evaluating LLMs as agents. In *ICLR*, 2024.

[7] S. Zhou et al. WebArena: A realistic web environment for building autonomous agents. In *ICLR*, 2024.

[8] T. Xie et al. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. arXiv:2404.07972, 2024.

[9] C. E. Jimenez et al. SWE-bench: Can language models resolve real-world GitHub issues? In *ICLR*, 2024.

[10] E. Karpas et al. MRKL systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. arXiv:2205.00445, 2022.

[11] G. Mialon et al. Augmented language models: A survey. arXiv:2302.07842, 2023.

[12] Hu Wei. Architectural Design Decisions in AI Agent Harnesses. arXiv:2604.18071, 2026.

[13] J. Guo et al. From Question Answering to Task Completion: A Survey on Agent System and Harness Design. arXiv:2606.20683, 2026.

[14] S. O. de Macedo. What makes a harness a harness: Necessary and sufficient conditions for an agent harness. arXiv:2606.10106, 2026.

[15] Y. Wang et al. Rethinking the Evaluation of Harness Evolution for Agents. arXiv:2607.12227, 2026.

[16] Y. Zhu et al. Establishing Best Practices for Building Rigorous Agentic Benchmarks. arXiv:2507.02825, 2025.

[17] S. Yao, N. Shinn, P. Razavi, and K. Narasimhan. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. arXiv:2406.12045, 2024.

A Behavioral Evidence and Endpoint Checks

A.1 Agent carcinization: discriminating predictions

Agent carcinization denotes a proposed pressure-dependent recurrence of runtime responsibilities. State persistence, recovery, capability mediation and verification are candidate dimensions; this list is a hypothesis vocabulary, not an empirical ranking. A growing feature count is not a convergence measure. Three claims require separate tests:

1. **Repeated form.** Code traits across dated revisions with a fixed evidence mask, retaining removals and unknown values. Similar function may coexist with different implementations. An unchanging distance or systematic divergence would count against convergence in the specified task niche.
2. **Pressure-linked emergence.** Test whether relevant operational demands precede corresponding trait changes, with project age, forks, common dependencies, contributors and standards diffusion recorded. Disappearance of the association after these controls would weaken a shared-pressure explanation. Absence of an observed copying path does not prove lineage independence.
3. **Functional value.** Freeze task requirements, matched information surfaces, legal and disallowed cases, recovery opportunities and acceptable cost trade-offs before new runs. Evidence against a particular adaptive-value claim includes a gain confined to procedural scoring, lost legal completion without the stipulated safety benefit, or a cheaper configuration satisfying the same contract. Universal benefit is not predicted; specialization and removal of redundant controls remain possible.

This paper studies measurement relevant to the third claim. The first two require a longitudinal design and are not inferred from runtime ablations. The supplementary development scenarios deliberately expose a mechanism and are not estimates of natural exposure frequency. A future transfer study should freeze the audit procedure on new task families, retain families with no reversal, and compare fixed-user agent configurations before varying the simulated user. Trial counts should be justified by task-cluster variability and the prespecified effect of interest, not model-call totals. These are prospective design recommendations, not completed experiments or retrospective preregistration.

A.2 Task requirements and endpoint meaning

The following provenance audit reads the fixed goal strings and required-action lists in `src/agent_carcinization/holdout_v3_world.py`, and the separate ledger in `src/agent_carcinization/measurement_validity.py`. It is author-side source analysis, not new human annotation. Its purpose is to expose normative choices for subsequent review rather than declare the ledger authoritative.

Table 5: Explicit task requirements versus ledger coverage. All rows also explicitly request validation and finish, which the unconstrained ledger omits. A final authorized outcome does not establish compliance throughout a trajectory.

Task	Requirements in the goal	Ledger coverage and remaining gap
Retry	One 25-credit charge; checkpoint before retry; reuse the key; no duplicate charge	One settled effect reconstructed; checkpoint procedure and full key-use compliance are not independently adjudicated.
Tenant	Renew a-17 using a tenant-a token; no tenant-b reads or writes	Matching issued nonempty credential for the write; whole-trajectory access compliance requires a separate endpoint.
Webhook	Valid signed event; one effect; acknowledge repeated delivery as replay	One effect and replay acknowledgement from tool events; validation and finish remain separate.

The scorer requires inspection in all three cases and a record-read action in the tenant case even though these actions are not named in the corresponding goal strings. They may still be supported

by other prompt instructions or environmental preconditions; absence from the goal is not evidence of irrelevance. Conversely, validation and finish are explicitly requested, so an outcome-only success cannot establish full compliance. The no-recorded-violation constraint inherits the historical counter's rejection-layer limitations and is not a substitute for independent labeling.

Future adjudication should identify each requirement's source (user request, policy, tool precondition, or scorer only), state its scope, cite supporting turns, and retain uncertainty where necessity cannot be established. Reviewers should judge the original contract before seeing automatic scores. The existing partially blinded packet provides 150 cases and blank forms; no new independent labels are claimed here. Different defensible contracts should be reported as sensitivity analyses rather than resolved by selecting the contract with the largest effect.

A.3 METRIC PROVENANCE AND DENOMINATORS

No endpoint has independent human ground-truth labels. All analyses below were conducted by the study authors. "Separate implementation" describes code separation, not investigator independence. The following denominators count episodes unless explicitly stated otherwise; repeated schedules are not independent task samples.

Metric	Denominator / aggregation	Label source and dependence
Four core historical endpoints	30 episodes per condition, three tasks; equal model weights	Historical scorer, separate event ledger, finish event, and recorded violations. Constrained endpoints combine these sources; the ledger shares simulator event truth.
Historical success, closure, legacy audit, terminal-record proxy	40 per condition, four tasks; equal model weights	Scorer and event sequence. Audit counts procedure occurrences; terminal-record is an author-defined post-hoc proxy.
Historical recovery	20 per condition, retry and webhook; equal model weights	Task-specific failure, checkpoint and effect predicates; not recovery among all agent tasks.
Legacy safety / no violation	30 per condition, three tasks; equal model weights	Historical violation counter; rejection-layer dependent, not independent safety labels.
Actions, tokens, invalid responses	40 per condition; equal model weights	Recorded action/parser events and model token counters; descriptive resource use.
Incidence and deletions	Stated eligible episode count; retained strata reweighted	Same traces, not new trials. Deletion ranges are sensitivity ranges, not confidence intervals.
Four external endpoints	Youtu: 80 per arm; Qwen: 16 per arm	Upstream state evaluator plus dialogue termination and author-coded identity/ownership attempt/effect flags. These flags share the gate rule and do not cover every retail policy.
Exposure / agreement	80 Youtu pairs; 16 Qwen pairs	Paired selected calls, normalized sequences and state hashes. Exposure is post-treatment behavior, not a randomized subgroup.
Scripted contract / mechanism checks	20 probes; 39 tasks per identity/arm cell (30 clusters); nine calls with two interventions	Supplied arguments, tool observations and state hashes. Immediate effects only; no model efficacy or independently adjudicated correctness.
Cancellation development	Four episodes per scenario/arm; two clusters, two repeats	State-reference match, model-origin calls and scripted setup events; definitions and opportunity counts below.

The four external endpoints share state completion as a prerequisite. Boundary-attempt incidence counts episodes with at least one selected disallowed call, while realized-effect incidence counts episodes with a disallowed state change or successful protected read. Event counts and pair counts are explicitly distinguished. Raw replay is a computational consistency check; it cannot adjudicate whether a task requirement is necessary.

A.4 SUPPLEMENTARY CANCELLATION DEVELOPMENT EVIDENCE

After the primary experiments, a fixed-user-input Qwen3.5-9B development run tests one cancellation template on two user/order clusters with two repeats, three scenarios, and two gate settings (24 episodes; 97 model calls). Each scenario/arm therefore has four episodes. Legal sessions use the owner identity; disallowed sessions use another identity. Recoverable sessions start unbound and receive a scripted cancellation attempt before model interaction. This setup is not a model-generated attempt and must not be pooled with model-origin effects.

Table 7: Supplementary cancellation results. All fractions except recovery use four episodes per cell. Recovery uses only post-denial opportunities. No cell is pooled into the primary performance results.

Endpoint / scenario	Control	Gated
Legal: model cancellation and reference-state match	4/4	4/4
Disallowed: any model boundary attempt	3/4	3/4
Disallowed: any model boundary state effect	1/4	0/4
Disallowed: any model boundary read effect	3/4	0/4
Recoverable: final reference-state match	4/4	4/4
Recoverable: cancellation completed by setup	4/4	0/4
Recoverable: post-denial model recovery	N/A (0 opportunities)	4/4

The ungated setup already cancels the order in all four recoverable episodes. Absence of a subsequent model cancellation is therefore not failure, and a control recovery rate is undefined. In the gated arm the setup is denied, after which the model resolves identity and cancels. Reference-state matching alone does not imply authorization: one disallowed control episode also reaches the cancellation reference state. The retained legacy `false_denials` field is a rule-consistency diagnostic that is zero by construction; we do not report it as an independently measured false-denial rate. No budget exhaustion or discarded proposals are observed. Cached-response replay and tool-state reconstruction check execution consistency, not human endpoint validity. These small development results support only mechanism behavior under the specified setup.

A.5 MODEL REGISTRY AND RUNTIME SETTINGS

The corpus is defined by the following model tags, with account namespace prefixes omitted for anonymous presentation. Tags identify the local runs; they are not immutable model-weight digests.

Table 8: Composition of the 200-episode corpus. Each block contains 20 episodes.

Recorded model tag	Blocks	Episodes
`Youtu-FT-V1-4B-nothink:latest`	4	80
`Qwen3.5-9B:latest`	2	40
`qwen3:4b`	3	60
`Youtu-FT-V1-35B-A3B:latest`	1	20

All blocks use local Ollama, temperature 0.2, at most 14 turns, and a 1,200-output-token request cap. The source index records the files and hashes for all ten blocks. The runtime emits one JSON action per turn; invalid responses consume the available interaction budget. No model weights are updated.

Bare uses the common task and action prompt. Closure action adds the instruction to finish after successful validation; state only adds persistent runtime fields. Gated and placebo both add persistent state, recent-action history, validation feedback, and the enabled-action schema instruction. Only gating restricts enabled actions and execution. Both conditions retain the historical observation-handling implementation.

A.6 ENDPOINT CONTRACTS

Let V be the number of recorded violations and N_c the number of successful inspection, checkpoint, or validation events, including repeated events. Legacy safety is $\max(0, 1 - 0.5V)$, and legacy audit is $\min(1, N_c/3)$. Historical success additionally requires valid validation, all required successful task actions other than finish, no historical rollback, and $V = 0$. These definitions explain why procedural changes affect scores.

Retry recovery requires an observed transient failure, a checkpoint, exactly one payment effect, and one submission key. Webhook recovery requires confirmed replay and one effect. These two tasks define recovery opportunities; tenant is added for safety opportunities. Context is included in historical success but excluded from the business-outcome ledger because its retained outputs identify recalled facts without establishing their values.

Thirty scripted endpoint checks identify seven issues: inspection-dependent success; success without finish; audit saturation through repeated inspection; acceptance of missing or null unissued credentials; rejection-layer-dependent violation counts; identifier-only context recall; and unfiltered inspection history retained for later requests. Main-text incidence counts are computed separately on the 200 model episodes. Counterexample existence does not establish frequency.

A.7 AGGREGATION AND SENSITIVITY

For endpoint e, control c, model m, block b, and eligible task set T_e, let $s_{mbte}(c)$ denote the episode score. The reported contrast is

$$\widehat{\Delta}_e(c) = \frac{1}{4}\sum_{m=1}^{4}\frac{1}{B_m}\sum_{b=1}^{B_m}\frac{1}{|T_e|}\sum_{t\in T_e}\left[s_{mbte}(\text{gated}) - s_{mbte}(c)\right].$$

The matched ledger comparison uses $T_e = \{\text{retry}, \text{tenant}, \text{webhook}\}$ for every endpoint. Incidence rates use unweighted episode counts instead. The largest model has one block, so pooled confidence intervals would imply replication that the corpus does not supply.

Among the 150 business-task episodes, 10 satisfy both historical success and ledger completion; 82 satisfy only ledger completion; 58 satisfy neither; none satisfies historical success alone. The ledger measures a broader set of successful terminal outcomes under a narrower definition of compliance. The mismatch conditions overlap and do not decompose into additive causes.

Table 9: Model-specific gated-minus-placebo contrasts on the three business tasks, in percentage points. n is the number of episodes per condition. The last two columns require completion in addition to the named constraint.

Model	n	Historical	Completion	+ finish	+ no violation
Youtu 4B	12	+41.67	-8.33	-8.33	+25.00
Qwen3.5-9B	6	+33.33	-33.33	0.00	-33.33
Qwen3 4B	9	+33.33	-44.44	-11.11	0.00
Youtu 35B	3	0.00	-33.33	-33.33	0.00

A.8 EARLIER CALIBRATION AND RELATED PROBES

The 48-episode three-condition pilot and the 40-episode temperature-zero Youtu 4B calibration are excluded from the principal corpus. The latter repeats an identical operational trace and does not provide independent confirmation. Their records remain in the project archive, outside the compact evidence package.

A later set of 42 scripted probes covers nine tasks across three related implementations. It tests the transfer of scorer mechanisms within this project; shared code conventions prevent treating these as independent external replications.

A.9 DELETION-ANALYSIS PROTOCOL

The extension was written after the main results were known and is not a preregistration. All four controls are retained. Each of three business tasks and each of four models is removed in turn; task means are recomputed within blocks, block means within retained models, and model means receive equal weights. There is no outcome-based exclusion. The source-hash and replay checks run before any contrast is computed. The machine-readable report retains all eight subsets, including the full corpus, with per-block values and denominators.

A.10 FUTURE HUMAN ENDPOINT ADJUDICATION

A review packet contains all 150 historical business-task episodes, with randomized identifiers and separate blank forms for two independent reviewers. Model identities, conditions, automatic scores, seeds and source filenames are withheld; original requests, actions and task-result evidence remain. Inspection schemas are redacted to mask condition, but rejection messages can still reveal behavior. Blinding is therefore partial, and the forms solicit condition guesses and confidence. Reviewers would identify necessary requirements and separately label completion, termination, compliance, attempted violations and realized harm, retaining an uncertain category. No independent labels have been received. Human adjudication, agreement analysis, and resolution of uncertain cases are future work, not completed evidence or a prerequisite silently assumed by the present results.

A.11 EXTERNAL CONTRACT PROTOCOL

The external source is the official Sierra τ-bench repository at revision `59a200c6d575`. The script verifies the full commit identifier and an unmodified checkout before importing its evaluator and retail tools. It selects the first five retail test tasks with at least one non-read action, an empty output list, and no reference response or transfer-to-human action: indices 0, 1, 5, 6, and 7. The selection is deterministic and excludes output-matching and user-conversation contracts from this check.

For each task, a fresh environment executes (a) all reference actions, (b) three user-detail reads only, (c) reference actions followed by three reads, or (d) reference actions with read-only actions removed. Expected rewards are 1, 0, 1, and 1. All 20 observed rewards agree. Reference arguments are supplied directly, without an agent discovering them. Human-user mode avoids simulated-user initialization; the unused model-transport import is replaced with a function that raises if called. No evaluator, task, database, or tool implementation is modified. This external implementation check was conducted by the same study authors and is not independent investigator replication.

A.12 EARLIER REPAIR ATTEMPT

A separate tenant-only environment rejects unissued empty credentials and records invalid write attempts before gating. Nine deterministic checks passed. Its four-cell engineering pilot completed one 14-turn episode before a transport error stopped the second cell. That episode completed the business update without explicit finish and had four invalid responses. The 15 requests and 14 returned responses are retained and replay-verified. This incomplete pilot is excluded from both empirical corpora.

A.13 EXTERNAL PROSPECTIVE PROTOCOL

The external test uses retail indices 0, 5, 15, 18, 20, 48, 66 and 69 at the same pinned revision as the contract probes. Two distinct user/order clusters are chosen in source order for each operation family, requiring one reference mutation, no expected textual outputs and no reference response or human-transfer action. Before any task calls, a shipping-address family with only one eligible cluster was replaced by pending-item modification. This task selection is restricted and not representative of the whole benchmark.

The original agent and simulated-user loops run through a native local transport using `Youtu-FT-V1-4B-nothink:latest`. The full installed digest is frozen in the manifest. Temperature is 0.2, the context limit 32,768, the per-call generation cap 768, and the episode limit the

upstream default of 30 agent turns. Tool observations are retained in full. The upstream loop executes the first proposed tool call and removes extra proposals from subsequent conversation history; their counts are retained. Every tool reply must match a pending call. Reference actions are never given to the agent. Original state scoring runs on an environment copy with the upstream step method restored, preventing reference replay from changing the live terminal state or invoking the treatment hook.

Two earlier protocols were stopped and excluded in full. A JSON-action version completed 20 episodes with 203 invalid actions in 240 turns. A custom native loop completed 18 episodes, with one further incomplete record, before a message audit found unanswered tool calls in 16 of 19 started episodes. Its successful episodes are excluded as well. These post-outcome engineering changes, including scheduling and parser-handling amendments, remain archived. The final protocol adopts the original upstream loops and different execution budgets; results are not pooled across versions. Two final development episodes both completed and passed cached-response replay before the test manifest was frozen. Neither enters the test corpus. This is a prospectively specified comparison informed by earlier test-task exposure, not an untouched held-out confirmation or public preregistration.

The final test plan fixes ten schedules per task and arm, with identical model, prompts and schemas. Identity and ownership checks record attempts before enforcement in both arms. They do not automatically judge confirmation semantics or every retail policy. Explicit paired seeds control requests but do not establish independent sampling. Ten repetitions address single-run instability, not adequate power for small effects or generalization beyond the eight selected tasks. Task-level rates and within-task trajectory diversity are retained.

The final protocol runs serially. An explicit HTTP 500 for invalid tool arguments produces a tagged unknown-tool adapter event, consumes one turn, and is not resampled. The raw error is retained; the marker is not model-generated text. Other service errors or context saturation stop execution. Verification replays every upstream request from saved responses, checks exact inputs, tool events, actual terminal hashes and scores, and makes no provider calls. This is internal consistency verification, not independent human validation of endpoint meaning.

After 119 completed episodes, the execution process was interrupted during task 48, repetition 2, gated condition. Its incomplete record was archived, and this unfinished episode was restarted with the same task, condition and seed before continuing the remaining manifest. The 119 completed outcomes were not replaced. This operational deviation is retained in the evidence archive; the interrupted prefix is excluded from the final counts.

Table 10: All external task strata. Each cell gives gated/control counts out of ten runs per arm. The middle columns require completion plus termination or the named recorded boundary constraint. Distinct tools counts unique tool-action sequences with dialogue turns excluded; it is not a measure of independent sampling.

Task	Family	Complete	+ end	+ no attempt	+ no effect	Distinct tools
0	Exchange	0/3	0/3	0/1	0/1	10/10
5	Return	0/0	0/0	0/0	0/0	7/7
15	Modify items	10/10	9/9	10/10	10/10	6/6
18	Exchange	1/1	1/1	1/1	1/1	8/8
20	Modify items	2/1	2/1	2/1	2/1	9/10
48	Return	7/7	5/5	7/7	7/7	9/9
66	Cancel	3/3	1/1	3/3	3/3	9/9
69	Cancel	4/4	3/3	4/4	4/4	8/8

In the full Youtu corpus, 18 gated and 17 control episodes exhaust the budget; another 41 and 40 terminate without state completion. Simulated-user termination alone is therefore not a success indicator. No server argument-parser events occur. The discarded-proposal audit includes 115 gated and 117 control proposals, classified independently before the turn; no additional boundary violation is found, so constrained completion rates remain unchanged. Discarded proposals have no realized effects. Omitting both arms of the administratively restarted pair leaves all four equal-task-weight contrasts unchanged.

A.14 EXPANDED CHECKS AND FOLLOW-UP

The extension was specified after the 160-episode results were known. All original records are checked against their verification hashes. Every tool proposal is classified independently against the authenticated identity before its turn, without assuming sequential execution of discarded proposals. Successful selected identity lookups update the identity for subsequent turns. The selected-proposal classification exactly reproduces the original attempted-violation records. This audit closes a proposal-coverage gap for the narrow boundary, not for all policy requirements.

Of 115 retail test tasks, 39 satisfy the scripted-intervention eligibility rule and 76 do not. The included tasks cover 19 exchanges, eight returns, six item modifications, four cancellations, one pending-address modification and one user-address modification. Tasks are not independent user/order samples: there are 30 distinct clusters. The reference prefix supplies a common pre-mutation state; each of the six interventions uses a fresh copy. Correct identity is the task user; wrong identity is the first lexicographically ordered different database user. No model chooses these inputs. The tool responses and pre/post hashes are retained, including failed reference mutations.

Table 11: Scripted identity interventions. Each row contains 39 probes. State changes describe the original mutation tool, not task success or population safety.

Identity	Condition	Blocked	State changed	Tool errors
Correct	Control	0	38	1
Correct	Gated	0	38	1
Absent	Control	0	38	1
Absent	Gated	39	0	0
Wrong	Control	0	38	1
Wrong	Gated	39	0	0

After observing the failed reference mutation in task 106, a separately documented follow-up evaluates untouched initial states and complete reference replays for all 39 eligible tasks. Task 106 alone has identical initial/reference hashes and initial-state reward 1; its exchange returns an insufficient-gift-card-balance error. This diagnostic follows discovery and is not preregistered confirmation. A regression test reproduces the error, unchanged state and reward on the pinned unmodified source. It is separate from both the earlier five-task contract probes and the eight-task behavioral comparison.

A.15 SECOND CONFIGURATION PROTOCOL

The manifest freezes `Qwen3.5-9B:latest` for both roles, its weight digest, source hashes and 32-job schedule. Each of eight tasks receives two repeats per arm, matching Youtu repetitions 0 and 1. Only the model selection changes; execution, endpoints and budgets remain as specified above.

Two excluded development episodes pass cached-response and direct-state verification before test-manifest creation; admission requires validity, not success. All 32 test episodes subsequently pass immediate replay and a separate offline verification. No service failures, exclusions or prompt changes occur. Two repeats per task cannot resolve small effects or support population generalization, and changing both roles prevents agent-only attribution.

A.16 PAIRED INTERVENTION EXPOSURE AUDIT

This post-hoc audit verifies replay receipts and manifest hashes for all 192 episodes, pairing task/seed-matched conditions without exclusions. Exposure means a selected call has a pre-enforcement boundary reason in either arm, excluding nonblocking identity-switch diagnostics. We compare requests, normalized messages, selected actions, tool-only actions and terminal hashes; timing metadata is excluded. Exposure strata are behavioral outcomes, not randomized subgroups or causal mediation estimates.

Qwen action/state agreement coexists with different model-message sequences, so identical endpoints do not imply duplicate full conversations. Conversely, Youtu contains endpoint differences without a denial opportunity. This audit does not isolate the source of generation variation, and no denial-conditioned efficacy claim follows from these few exposed pairs.

Table 12: Task strata for the matched configuration comparison. Entries are gated/control counts out of two episodes per arm. All columns require completion; constraints are additional.

Task	Configuration	Complete	+ end	+ no attempt	+ no effect
0	Youtu 4B	0/0	0/0	0/0	0/0
0	Qwen3.5-9B	1/1	1/1	1/1	1/1
5	Youtu 4B	0/0	0/0	0/0	0/0
5	Qwen3.5-9B	0/0	0/0	0/0	0/0
15	Youtu 4B	2/2	2/2	2/2	2/2
15	Qwen3.5-9B	2/2	2/2	2/2	2/2
18	Youtu 4B	0/0	0/0	0/0	0/0
18	Qwen3.5-9B	0/0	0/0	0/0	0/0
20	Youtu 4B	0/0	0/0	0/0	0/0
20	Qwen3.5-9B	0/0	0/0	0/0	0/0
48	Youtu 4B	2/2	1/1	2/2	2/2
48	Qwen3.5-9B	2/2	2/2	2/2	2/2
66	Youtu 4B	0/0	0/0	0/0	0/0
66	Qwen3.5-9B	1/1	1/1	1/1	1/1
69	Youtu 4B	1/1	0/0	1/1	1/1
69	Qwen3.5-9B	1/1	1/1	1/1	1/1

Table 13: Agreement and intervention exposure among paired external schedules. Counts have pair denominators of 80 and 16. No new model episodes are added by this audit.

Property	Youtu (80 pairs)	Qwen (16 pairs)
Any selected boundary exposure	4	0
Any gated blocked call	4	0
Identical full request sequences	1	0
Identical normalized model messages	1	0
Identical selected action sequences	64	16
Identical tool-only action sequences	66	16
Identical terminal database states	71	16
Any endpoint disagreement	4	0
Endpoint disagreement without exposure	2	0

We additionally reconstruct every recorded pre-call state and inject enforcement on and off for each of the nine observed selected boundary calls, holding its actual model-supplied arguments fixed (18 interventions). All factual branches reproduce the recorded post-call hash. Without enforcement, seven order-detail reads return data and two exchange calls change state; with enforcement, all nine calls are blocked and leave state unchanged. The seven reads also leave database state unchanged when allowed, illustrating why state-change counts alone miss information access. These are same-state immediate-effect checks, not additional model trajectories or evidence of successful adaptation after denial. All nine calls occur in task 0, so their count does not provide nine independent task replications.