LONG STUDY

Receiver Reliance

A structural handoff mechanism, its evaluation, and its limits

James Dimachkie · Solo-led research with substantial AI assistance
Evidence through September 5, 2026 Pacific time · Research paper

Archive: 10.5281/zenodo.22492561.

Abstract

Receiver Reliance (RR) checks whether evidence supports a particular receiver's use of an exact work product. We evaluated its full operational procedure in synthetic multi-agent arithmetic and item-selection workflows. The study comprised 1,805 episodes, including 24 semantic controls. Every episode measurement was valid. Relative to transparent release and two fixed static policies, RR reduced currently improper receiver presentations by estimated means of 1.00, 1.00 and 0.75 per seeded episode; all three conditional simultaneous intervals excluded zero. This reduction accompanied lower seeded-task completion: 80/162 under RR versus 150/151, 154/155 and 139/157 under the alternatives. Clean-completion effects remained uncertain, and every arm failed all six semantic controls despite completing their trajectories. RR enforced the tested structural requirements. Field utility, semantic correctness and advantage over an equally expressive conventional policy remain unestablished. Earlier studies found conventional parity and no demonstrated benefit from the tested learning and prompt interventions.

Key takeaways

What RR does

- Checks whether evidence supports a receiver's specified use of an exact work product.
- Enforces the declared structural requirements before the host releases that input.
- Reduced improper presentations in the tested synthetic conditions, with all three seeded comparison intervals below zero.

What RR does not establish

- Correctness of an answer merely because its evidence relationships are valid: every arm scored 0/6 on semantic controls.
- Better overall utility: RR correctly completed 80/162 seeded tasks, and clean-completion effects remained uncertain.
- Field efficacy or advantage over an equally expressive conventional policy; that comparator was not tested.

When to consider it

- Consider an evaluated receiving policy when a workflow has explicit, observable evidence/use requirements and the consequences of violations and holds can be assessed.

Need to verify answer content?	→ Use a defined semantic checker; RR alone is insufficient.
Required facts unavailable?	→ Improve observations before relying on the check.
Violation and hold costs unknown?	→ Assess costs prospectively before deployment.
Requirements and costs explicit?	→ Evaluate against an equally capable conventional policy.

Figure 1. Decision guide for considering an RR-based workflow. These are application questions, not validated deployment criteria.

Contents

1. The question at a receiving boundary

An agent receives a recommendation, a calculation or an intermediate plan from another agent. Before acting, it needs to know what that record supports. The answer can depend on the record's identity, the evidence attached to it, the authority behind a declaration and the specific operation the receiver intends to perform. None of those properties alone guarantees that the content is correct.

RR began with the concern that errors acquire apparent authority as they move through delegated work. A fluent downstream answer may conceal that the receiver relied on an obsolete revision, evidence about a different object or a declaration that did not cover its use. RR puts an explicit decision at that receiving edge. Its scope is the specified relationship between object, evidence and use, rather than an unrestricted judgment that the record is trustworthy.

Consider a parcel-allocation workflow. One agent creates recommendations, another groups them into a manifest, and a third produces the final allocation. Evidence about the recommendations does not automatically apply to the new manifest. Even evidence that does apply cannot make an incorrect grouping correct. Further, executing the supplied plan differs from reading its facts and recomputing an allocation. A record may be unsuitable for the first operation but usable for the second.

This distinction became observable in the earlier operational experiments: reference identities could survive while task scope was lost, and a final recomputation could recover from an incorrect intermediate grouping. A universal label attached to the whole record would hide both facts. The practical question is what a particular receiving operation requires and whether those requirements can be checked from available observations.

Specifying the relation, computing it from observations and measuring its effects require different evidence. The newest experiment tests the implementation and measures exposure and task completion against three comparators.

What this paper contains

Sections 2-6 explain the mechanism and completed local experiment. Section 7 retains the earlier research results that constrain interpretation. Sections 8-9 position the contribution and provide reproduction routes. The short study is the entry point for readers primarily interested in the result. The preserved 50-page account contains additional chronology, source locators and development corrections.

Sources: unabridged account, earlier operational results.

2. The mechanism and its boundary

The reference engine accepts a host-assembled fact profile and applies a fixed predicate order. Its classes distinguish malformed or boundary failures, binding or conflict failures, omissions or incompleteness, and a residual valid class. These are decisions under a declared profile. A correct implementation does not establish that the profile faithfully describes the outside world.

The full operational package adds the receiving relations around that engine. At each registered receiving node it performs shared canonical-byte and closed-schema admission, checks whether the raw facts support the required derivation and identities, invokes the pinned reference engine through the specified constructor, classifies lineage, and classifies acceptance. Lineage covers the receiver, parents, route, slot, carrier, generation, revocation, fence and validity interval. Acceptance covers direct or one-hop standing, lifecycle, source completeness, snapshot age and exact intended use.

The outputs PASS, BLOCK and DEFER have their declared meanings and a common remediation interface. The host captures observations, creates the input, invokes a fresh decision process and releases the captured input to the model only after PASS. It retains the engine-call and witness-construction evidence. The treatment does not decide whether the model's task answer is true.

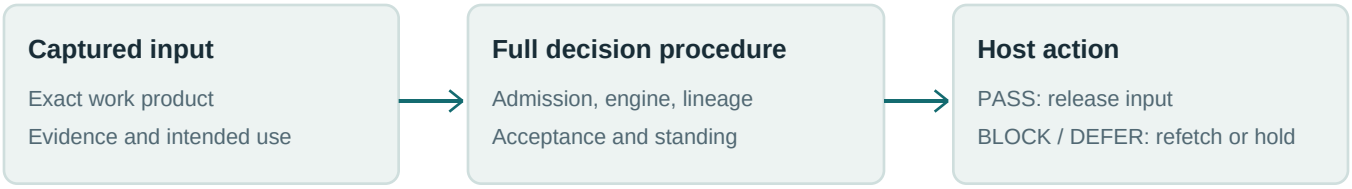


Figure 2. RR checks the structural relationship at the receiving boundary. The host supplies observations and enforces the result. Task correctness is evaluated separately.

What the host must supply

The broader contract separates the check from responsibilities such as authentic observation, binding a decision to the actual input and use, enforcement, appropriate replay handling and the relationship between a check and its effect. A self-verifiable hash establishes an identity relation under a specified encoding. It is not a digital signature or evidence that the source was authorized to assert a fact.

This experiment used an honest controller to construct and record local administrative events. Its custody checks make those executions inspectable within that boundary. They do not establish administrator-resistant attestation, deployment security or the authenticity of independently supplied external facts. A decision procedure cannot recover facts that its observation system never captured.

Which version was tested

RR has separate histories for its foundations, engine generations, software releases and study authorities. Their version numbers are not a single maturity scale. The local successor restored the operational treatment-v6 interface and determinate semantics, with explicit resolutions for conversion, identity namespaces, parsing and other ambiguities. Two independently authored treatment implementations were reconciled on development inputs; package A was fixed for collection.

Full operational treatment denotes the entire declared decision procedure. The four seeded conditions test selected failures within that procedure.

Sources: full treatment and disclosed resolutions, original local admission, historical technical paper.

3. Experimental design

Tasks, graphs and populations

Each episode began with a source model and then used one receiver, a chain of three receivers, or a fork/join with four receivers. Two tasks required summing integer groups or selecting item identifiers by grade and count. Receivers combined actual parent outputs with fresh task groups. The fork/join instructions preserved overlap correctly. A standalone scorer calculated ground truth from the original task groups; models received neither gold answers nor another arm's response.

The evaluation reservoir contained 256 worlds per task/graph cell and was separate from development. A fixed arm-neutral function assigned a decoding seed to each world. Each arm drew its worlds uniformly with replacement. A separately drawn shuffle fixed the full execution order. Repeated worlds remained isolated repeated intervention executions, not newly discovered natural incidents.

Four equally weighted seed conditions introduced an administrative defect at the first receiver: no adoption declaration, an adoption naming the wrong use, expiry of a local ingress grant before a benign file copy, or a parent envelope delivered at a different revision with its payload preserved. The host created the event before capturing the resulting facts. It did not rewrite a completed observation to fit a label. Later receivers had current-use records of their own, so a valid child declaration could clear an earlier administrative defect.

Two equally weighted clean conditions supplied direct or one-hop delegated standing. A separate semantic control changed the source answer after retaining its genuine request and response: add one to a sum or toggle an item identifier. This was a declared constructed input, not a falsified account of the model's output. Its final correctness still had to be measured; a downstream model could in principle recover.

Four packages under one host

Arm	Declared treatment	Interpretation
U	Shared admission followed by transparent PASS	Reference for release without the additional decision stages
S4	Original independent static comparator, threshold four	Fixed static package
S1	Same static comparator, threshold one	Secondary static package
R	Complete operational R-all procedure	Full local treatment, with disclosed specification resolutions

The static comparator used three component and thirteen extended bits. Its thresholds were declared rather than tuned to the observed evaluation outcomes. Neither static arm implemented the complete dynamic relation in R-all. A conventional implementation that did so was not a fifth arm. This limits any claim that RR has a distinctive advantage over conventional policy as a category.

Every arm ran a real model trajectory. The common host applied the same capture, release, correction and resource rules to the same captured input and facts. Later trajectories could differ because of earlier decisions or different actual model outputs. Equal host rules therefore did not imply identical later inputs, and arm identity was excluded from common input identities.

Administrative repair and resource limits

A first non-PASS allowed one administrative refetch, preserving parent payloads and intended use. Even-indexed reservoir worlds had a correcting service; odd-indexed worlds retained the defect. The reservoir mixture was exactly half each, while the sampled mixture could differ. Refetch did not consult gold, regenerate an answer or repair task content. A second non-PASS held the receiver. Dependents without a required output were cancelled, while an available independent sibling could continue. A receiver was presented at most once.

The runtime used Ollama 0.33.3 with a pinned Qwen3 14B model on a qualified Colab A100-SXM4-40GB image. Requests used temperature zero, context length 8,192, JSON mode, thinking enabled and a 1,536-token generation cap. Each model operation had one HTTP attempt and a 180-second transport timeout. The episode deadline was 1,200 seconds, checked between operations and after the final operation. An observed late final answer could not count as timely completion.

These limits define the measured package. Malformed and token-limited outputs remained observed outcomes under the one-attempt rule. Collection itself retained a 14-hour active-time cap and a ceiling of three interruption resumptions. No VM-loss resumption occurred in the completed stage.

Endpoints and measurement failures

The primary seed endpoint, Y, counted actual receiver presentations satisfying at least one currently evidenced registered structural predicate. A presentation counted once even if multiple predicates held. The support ranged from zero to one, three or four according to graph. Every current presentation was checked, including newly arising downstream occurrences. An ancestor's label alone did not make a later presentation defective; descendants of an observed defective presentation were a separate descriptive quantity.

The primary clean endpoint required all receiver outputs to parse under the fixed strict grammar, the final answer to match independently calculated truth, and the episode to finish within its deadline. Seed completion, semantic errors, holds, retries, model calls, tokens and elapsed time were retained as separate descriptive outcomes. Semantic correctness meant exact agreement with the arithmetic or item-selection oracle over original groups. The scorer used deterministic rules; the earlier prompt pilot used qualitative AI grading, with uncertainty discussed in Section 7.

A fully observed hold, malformed model answer or known deadline failure remained an assigned outcome. Missing model or witness evidence was a measurement failure, not a favorable zero. A missing required observation invalidated each contrast depending on the affected slot; a global source or runtime identity failure invalidated the estimates globally. This law differs from the historical program's worst-case completion envelopes.

Sources: protocol, statistical specification, frozen schedule.

4. Qualification and analysis

Development qualification

Before evaluation, independent host review exposed three consequential problems. Identical payloads could collapse distinct parent identities; arm identity could contaminate common episode identity; and a sum instruction allowed a reasonable but unintended reading of how often to add a parent subtotal. The host gained node-specific canonical envelopes, common identities were corrected, and the prompt was clarified. The earlier calibration was stopped and preserved as invalid development evidence for capability selection.

A fresh development screen used a declared first-eligible selection order. The 8B nonthinking profile produced 3/24 correct and 23/24 fully parsed episodes; the 8B reasoning profile produced 15/24 of both; the 14B reasoning profile produced 24/24 of both and was selected. These are calibration results on constructed tasks, not a general model ranking. The qualified GPU run retained the selected model digest and decoding profile. Its separate 24-world development run was fully correct and parsed, with 88 model calls and a median episode duration near 19 seconds, versus about 152 seconds locally.

Treatment qualification covered 62 normative R cases and three comparator checks on both implementations, yielding 130 expected outputs; plain and observed executions agreed. Twelve wire fixtures exercised a disclosed parsing-limit resolution. A task-message interpreter then ran 264 synthetic host episodes and 714 A/B decision comparisons with fresh decision workers. That interpreter was a qualification stub, not LLM

efficacy evidence. Sixteen further live development episodes plus four repeated warm requests retained 52 model requests/responses and 44 A/B comparisons.

An independently authored scorer reconstructed task truth and current predicates without importing the treatment, host or workload code. Its checks used 282 retained development/qualification episodes and eight sensitivity families. Changing a supplied label or summary did not change the score; missing or spliced witness evidence failed measurement. Deadline fixtures covered a final answer arriving after the cap. Final collection review also required repairs to isolated observation-failure handling, interrupted-time accounting and exported invocation-finish evidence. Those were checked before exact-state admission.

Development and evaluation remained separate: candidate repairs preceded the study freeze, and all calibration and qualification attempts retained their original status outside the evaluation denominator.

Sample and precision

The study included 1,805 episodes across the four arms. The sample met the selected maximum interval half-widths of 0.6 current improper presentations per seed episode and 15 percentage points for clean completion. These targets specified the resolution of the estimates; they were not thresholds for significance or acceptable task loss.

Population	U	S4	S1	R
Seed	151	155	157	162
Clean	299	291	274	292
Semantic controls	6	6	6	6

The 1,805-episode population was fixed after collection began and before endpoint analysis, using the frozen schedule's first 1,800 entries plus five remaining controls. The selection record documents the outcome-blind assessment and the 619 deferred assignments from the original 2,424-slot schedule. All acquired observations and all 24 controls were retained; treatment, model and scorer remained unchanged.

Estimates and uncertainty

The episode was the inferential unit. Within each primary population, the predefined condition/task/graph cells retained equal weight despite unequal realized cell counts. The six fixed contrasts were R minus U, S4 and S1 for seeded Y and clean correct completion. Pooled arm means describe a different realized mixture and were not substituted for the stratified estimates.

For each contrast, the estimate is the weighted sum of within-cell mean differences. Its variance-bound term is S = sum over cells of w squared times L squared times (1/n_R + 1/n_comparator), where w is the cell weight, L the endpoint's support range and n the actual cell counts. The squared interval radius was (25/8)S. Square roots were rounded outward to six decimals, and intervals were intersected with possible contrast supports.

An exact finite exponential certificate bounds each directional tail below 1/512. The analysis law budgets for six intervals at each of two possible fixed reporting stages. A union bound over the 24 directional tails gives simultaneous coverage of at least 61/64, or 95.3125%. Only the present stage was executed. This coverage remains conditional on the declared sampling and runtime assumptions.

The conditional interpretation assumes stable seeded potential outcomes, or independent stochastic executions with the same conditional distribution. Fresh messages and processes, fixed decoding seeds and a small successful repetition check did not prove stationarity. The realized maximum radii were approximately 0.599 structural presentations and 14.93 percentage points of clean completion, meeting the selected precision targets.

Custody and analysis

A durable local lease identified each batch before execution. Each checkpoint was downloaded and verified against its emitted hash, complete manifest, assignments, closed invocation and byte-identical prior overlap before the next lease. Interrupted slots could not be rerun. The completed stage retained nineteen locally verified checkpoint archives and 88,865 unique evidence files.

All 1,805 episodes were scored VALID, with no global identity errors; all six contrasts were VALID_CONDITIONAL. This describes measurement integrity under the frozen checks, not universal software correctness or task success. An internal reviewer independently recomputed all six estimates, exact squared radii and outward bounds from episode scores without importing the production analyzer or estimator. Twelve predetermined traces also passed checks of assignments, witnesses and actual model-input correspondence. That sample was not a second complete trace census.

Sources: qualification, original admission, exact report, original-schedule accounting.

5. Results

Structural release

R minus comparator	Seed Y difference	Conditional simultaneous interval
U	-1.000	[-1.599, -0.401]
S4	-1.000	[-1.596, -0.404]
S1	-0.750	[-1.334, -0.166]

The equal-cell seed means were zero for R, one for U and S4, and three quarters for S1. No observed R seed episode presented a receiver while a registered current structural predicate held. U and S4 each retained one such presentation per episode. S1 removed the occurrence in one of the four equally weighted seed conditions. All three upper interval bounds were below zero, supporting a package effect on the constructed structural endpoint.

These values are counts per episode, not percentages or natural incident rates. The table rounds only for display. Exact estimates and outward interval bounds remain in the machine-readable report.

Clean completion

R minus comparator	Clean completion difference (percentage points)	Conditional simultaneous interval
U	-0.375	[-14.973, +14.224]
S4	+0.888	[-13.831, +15.607]
S1	-0.429	[-15.362, +14.505]

Clean-completion effects remained uncertain. The intervals include benefit and harm and establish neither equivalence, noninferiority nor demonstrated utility gain. No acceptable-loss margin was tested. The contrast plot shows the uncertainty around the small point estimates.

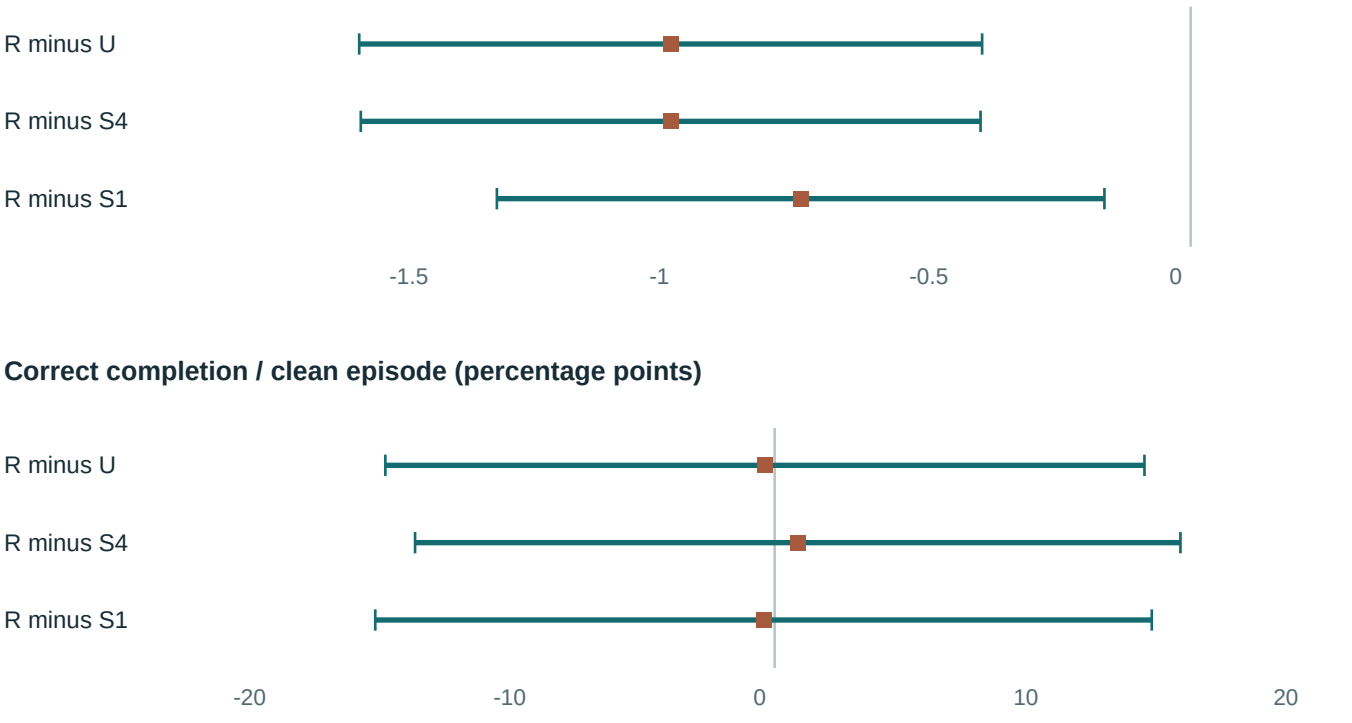


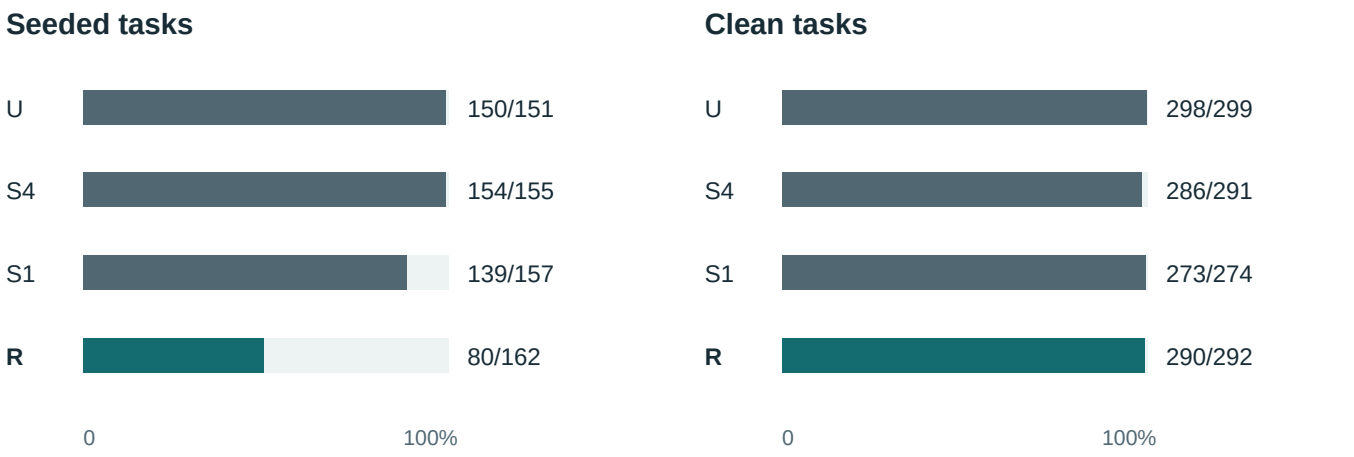
Figure 3. Six fixed initial-stage contrasts. Seed intervals are below zero; clean intervals cross zero. The two-report certificate gives joint conditional coverage of at least 95.3125%. Panels use different units. These intervals do not certify seed completion or semantic benefit.

The seed completion tradeoff

Arm	Correct seed completions	Correct clean completions	Seed retries
U	150/151	298/299	0
S4	154/155	286/291	0
S1	139/157	273/274	36
R	80/162	290/292	162

These are pooled descriptive counts over realized draws. R's 80/162 seed completions accompanied 81 held receivers. A correcting service could repair a relationship without changing the answer. A persistent defect stopped required work even when its payload remained mathematically usable. Lower structural exposure therefore coexisted with substantially lower completion.

Seed completion is reported descriptively because it was outside the six-contrast inferential family. All assigned outcomes, including known holds and token-limited malformed outputs, remain in the counts.



All four arms: 0/6 correct final answers in semantic controls.

Figure 4. Correct completion, with a common 0-100% scale. Counts are descriptive over realized draws, not the equal-cell primary estimates. Lower seeded exposure under RR accompanied fewer completed tasks.

Semantic controls

All 24 control trajectories completed. Each arm obtained zero correct final answers out of six. Its controls had sixteen semantically erroneous receiver inputs and sixteen erroneous receiver outputs, while the registered structural-error presentation count was zero.

Receivers were instructed to combine actual parent answers with their own new groups. They were not given the original source group needed to recompute the source answer; item-selection receivers were explicitly told

to retain parent IDs. A model could therefore follow its local instruction correctly while carrying the constructed source error into the final result. The scorer instead compared with truth from the original groups. This is a test of error propagation under the assigned decomposition, not a test of recovery with full source access.

R-all provided no semantic repair. Whether source access, independent recomputation or another intervention could recover the correct answer requires a separate comparison; these controls do not establish a general model capability ceiling.

Execution cost

Population and arm	Model calls	Mean episode seconds
Seed U	565	18.41
Seed S4	561	18.07
Seed S1	534	16.74
Seed R	395	12.38
Clean U	1,109	18.73
Clean S4	1,070	18.37
Clean S1	1,009	18.50
Clean R	1,064	18.42
All semantic controls	88	Arm means 18.17-20.25

Across populations, 6,395 actual model calls were retained. Clean episode means were similar descriptively on this runtime. R's lower seed time and call count partly reflect fewer executed receivers, so they are not equal-quality efficiency. Tokens and model wall time remain available in the report. No general latency distribution, monetary saving or deployment-cost advantage was estimated.

Sources: exact aggregate results, episode scores, receiver-message construction, independent scorer.

6. Interpretation

The host retained the inputs released to downstream models, and a separate scorer checked current structural predicates at each receiver. RR reduced exposure across four constructed seed families, two tasks and three graph shapes under one qualified model/runtime profile and an honest local controller.

The comparison estimates the effect of the full package without isolating individual components. The static policies did not express the complete dynamic relation. A conventional implementation of that relation could potentially make the same decisions; it was not tested here. Earlier conventional-parity findings concerned a narrower relation.

The value of enforcing a structural requirement depends on the consequences of violating it and of withholding legitimate work. This experiment measured release and completion; application-specific costs would determine the preferred tradeoff. Its clean-task intervals leave meaningful uncertainty about a completion penalty.

Testing field utility would require an authentic workflow, a capable conventional comparator and explicit costs for violations, repair, delay and legitimate work stopped, accounting for natural error prevalence and stakeholder priorities.

7. Earlier results and research context

Earlier studies tested different mechanisms and interventions; their counts cannot be pooled with the full-treatment comparison. The findings below provide context for that result. Detailed history remains in the unabridged account.

From vocabulary to inspectable contracts

Early foundations distinguished evidential support, permission and interpretation. That was useful because a well-supported statement need not authorize an action, and an authorized action can still be mistaken. The design's corresponding weakness was that required semantic fields could remain supplied judgments. A typed representation did not discover the meaning of arbitrary language or make an unsupported field trustworthy.

Later work bound evidence to the exact canonical delivered envelope and made applicability sensitive to intended use. Transformation created a new subject. A subject-binding constructor deliberately granted no new acceptance or authority. Independent review exposed an encoding ambiguity, resolved by specifying JSON escaping, number representation and the absence of a byte-order mark or trailing newline. The problem was specifying the object being hashed, not a failure of hashing itself.

A finite information audit showed that a deterministic classifier cannot distinguish cases with different labels when its observed bytes are identical. On fourteen development worlds, a task-content-only view merged two differently labelled pairs, forcing at least two exact-label errors. The full supplied representation retained those distinctions. This is a finite representational result, not proof that the labels were true or that the complete view is universally sufficient.

The direct-applicability relation distinguished APPLICABLE, INAPPLICABLE and UNKNOWN. If the required identity, use and time facts hold and no withdrawal is known, an incomplete withdrawal history still leaves two possibilities: a relevant withdrawal exists, or it does not. UNKNOWN preserves that uncertainty. A closed-history assertion comes from the host; the evaluator cannot infer its truth from a digest. Ordinary joins, equality and interval filters, an anti-join for withdrawals and a closure rule implement the same relation.

A separate coverage check compared the original required object identifiers with those actually present. It detected omissions and extras only if the original requirements came from an independently grounded source. Deriving requirements from an already incomplete intermediate record would certify the omission. Neither coverage nor applicability established answer correctness.

Finite checks included 110,592 applicability comparisons, 128 refinements and 2,912 coverage checks. The two applicability methods shared validation, formatting and a contributor; they were not independently authored complete RR implementations. These checks supplemented conditional arguments rather than establishing a new security primitive or authenticating host facts.

Replay and conventional comparison

The retained retrospective replay contained 408 records: 390 clean real-labelled, five defective real-labelled and thirteen seeded records. This corrected the earlier account of 398 real and ten seeded records. Origin and semantic labels remained incompletely verified. The sole additional defect detected by RR was seeded, and the comparator omitted an ordinary cross-record join. The replay therefore did not establish advantage over equally capable ordinary verification.

False-hold and stale-filename analyses reinforced a related lesson. An old filename does not itself make a proposition obsolete: unchanged clauses or historical uses can remain legitimate. A calibrated reduction in one hold counter cannot be equated with improved downstream utility. The proposition, intended use and outcome need their own evidence. The August precursor's twenty agent runs and 160 answers were exploratory, with uneven cohorts, rather than a clean confirmatory comparison.

Two narrow operational versions

On twelve constructed blocks, an initial operational version produced 3/12 correct outcomes under unchecked release, 2/12 under a conventional package and 2/12 under narrow RR. After joint refinements, the same population yielded 11/12, 12/12 and 12/12 correct outcomes. The refined totals included three correct explicit withdrawal abstentions per arm: unchecked release had eight productive correct allocations, and conventional and RR each had nine. The conventional implementation matched RR in both versions. The 240 retained calls show development improvement on reused cases, not an independent new-population efficacy estimate.

Transcript inspection found that reference identities could survive while scope was lost, and that a correct final recomputation could recover from an incorrect intermediate grouping. This is why exact identity, applicable evidence, retained task requirements and correct answers must be assessed separately. A simplified relation can be useful without justifying a distinct engine when ordinary composition produces the same decisions.

Learning and the prompt pilot

In a separate 160-state control model, ten learned policies were evaluated exactly after 100,000 training episodes and 131,346 decisions. Every final learned seed underperformed a simple heuristic. Exact correct-completion probability was 0.265234 for the planner, 0.258203 for the heuristic and 0.196914 for the learned mean. Lower learned resource use accompanied greater abstention. One retained state showed a policy abstaining because relevant terminal execution had not been sampled and all action values tied at zero. A safe-looking final policy did not prove learned safety. Full old training trajectories were not retained. This diagnoses the tested learner in its toy environment; it does not rule out reinforcement learning, other representations or better training methods for reliance decisions.

A finite structured-reporting contract later achieved conventional parity on twelve paired cases. A distinct prompt pilot tested an added evidence-use review procedure against a similarly sized generic review. It used six development and eighteen evaluation families, with 144 calls across the retained work. In the primary paired evaluation, the evidence-use procedure passed 6/18 families and general review passed 9/18. Five passed both, one only the added procedure, four only general review and eight neither. The estimate was -16.7 percentage points, with a conservative interval of [-50.9, +25.4] and exact paired p = .375.

The interval supports no demonstrated benefit, not proof of no effect. Constructed, potentially dependent families limit its sampling interpretation. Grading was also uncertain: initial readings differed on at least one criterion in 59/108 evaluation records, although only three whole-task outcomes differed. Under favorable readings, the evidence-use procedure passed six or seven of eighteen families and general review passed nine. All reviewers were AI systems within the same operator's project.

Formal results and interface failures

The formal record reported 872 properties: 766 proved, 105 bounded and one refuted. Atomic-predicate work reported 40,907,363 evaluations and 3,960 sealed pipeline calls but excluded surrounding behavior such as wire parsing, receipts, transcripts and wrapper rederivation. A second complete implementation separately had 592 full-surface differences across five mechanisms. These are compatible observations at different layers. Predicate agreement cannot certify an entire interface or the host applying its result.

Development of the observation instrument also exposed a more basic risk: an experiment could capture or reconstruct different bytes from those actually used while still reporting an apparently coherent result. The local successor's actual-release and witness checks address that specific boundary. They do not retroactively qualify the historical apparatus.

Historical study design

The historical public/adversarial program never reached its intended registration or execution. Its first exact row law implied 5,126,400 confirmation episodes under an incomplete whole-design closure. Later proposals of 10,448 and 3,072 episodes reduced cost but left unsupported margins, omitted gates or resource-selected precision choices. An affordable count did not establish that the intended claim could be answered. Reviews also corrected intermediate arithmetic, illustrating that a review conclusion does not make every calculation authoritative.

Runtime qualification remained difficult: an 8B model differed after cold loads, and CPU/GPU placements could differ despite deterministic settings. A 14B candidate's 200 matching pairs over 190 distinct prompt strings were bounded evidence, not qualification at proposed study scale. A separate design evaluator's synthetic N=896 example demonstrated arithmetic machinery, not a selected population or an admitted replacement study.

The historical program retained unresolved qualification, instrument, comparator, statistics and adaptive-diagnostic requirements. The newer honest-controller study adopted its own bounded scope and admission. The completed local study is separate from that broader program.

Sources: history and corrections, operational results, learning findings, prompt results and grading, technical and formal record, contribution validation.

8. Contribution, prior art and limitations

RR's broader architectural language overlaps established work. Proof-carrying authentication develops authorization from proofs; RATS separates evidence appraisal from a relying party's decision; policy-carrying data connects information and restrictions on use; and CaMeL separates trusted control from untrusted data with out-of-band enforcement. RR differs in mechanism and target, but a receiving-boundary diagram alone does not establish novelty. The retained field review also discusses AuthMem, MemTX and context-to-execution integrity without claiming to reproduce their performance.

Proposed pivots toward maintaining documents or providing research assistance encountered further prior art. FRUIT studies updating existing text from new evidence while preserving unchanged content; WiNELL addresses ongoing fine-grained updates. An initial assessment missed close work and was corrected. The archive supports specific implementation and evaluation lessons, not a claim that these broader research directions were unoccupied.

The project produced explicit object/use contracts, conditional mathematical results, executable mechanisms and empirical evidence. The full operational study establishes reduced structural exposure against its declared comparators. Earlier parity findings, counterexamples and measurement corrections provide additional material for implementation and evaluation.

Generalization is limited by synthetic tasks and administrative conditions, sampling with replacement from a finite reservoir, a single model/runtime profile, an honest controller, and coarse conditional intervals. Natural incident prevention, component effects, external authenticity and field utility remain outside the tested claim.

James Dimachkie directed the project with substantial AI assistance in specification, implementation, mathematical analysis, experiment execution, review and writing. Separated reviewers sometimes used different providers or contexts, but remained within one operator's project. These checks are internal review, not independent human replication.

Further research

The local study is complete on its stated terms. The historical program's 5,126,400-episode calculation belonged to a different claim and conservative design law; it is not a required next step or a universal lower bound on useful RR research. A successor can pursue a narrower question with its own justified comparator, outcomes and observation requirements.

Open question	Required comparison or evidence
Does RR add value over an equally expressive policy?	Implement and qualify that comparator on the same observations and host capabilities
Are prevented violations worth the work stopped?	Choose an authentic workflow and prospectively justify violation, repair and completion outcomes
How can semantic errors be detected or repaired?	Specify what source evidence a receiver can access and compare a defined intervention against a capable baseline
Will another team reproduce the result?	Obtain the exact local evidence and pinned environment, audit the scorer, and independently reproduce or replicate the relevant claim

The contracts, frozen workload, analysis and counterexamples are listed in the evidence guide below. Resource estimates for further work depend on the chosen question, design and runtime.

Sources: retained field assessment, novelty corrections and access limits. These are the dated source reviews used in the unabridged account; this editorial revision does not present a new literature search.

9. Evidence and reproduction

The editions are different reading levels of one research record. The one-page abstract states the idea and principal result. The short study explains the experiment and interpretation. This long study retains the core methods and earlier findings. The unabridged 50-page account and original 41-page paper remain unchanged, as do the frozen experimental authorities and observations. No public preregistration preceded collection. Publication of the completed record does not change the documented chronology of its local freeze and post-start amendment.

Evidence needed	Retained entry point
Exact contrasts, descriptive metrics and validity	analysis/initial-stage-20260906/report.json
All selected episode scores	analysis/initial-stage-20260906/episode-scores.json
Complete assignment accounting	analysis/initial-stage-20260906/original-schedule-accounting.json

Evidence needed	Retained entry point
Original specification and admission	TREATMENT_SPEC.md; PROTOCOL.md; STATISTICS_SPEC.md; freeze/admission.json
Sample definition and analysis law	stage_one/AMENDMENT.md; membership.json; admission.json
Retained raw checkpoint custody	collection-local/receipts/ and reproduction/artifacts/
Unabridged narrative and verification	paper/BEFORE_THE_NEXT_AGENT_ACTS.md; paper/VERIFICATION.json

Paths in this table are relative to study/2026-1/full-treatment; filenames grouped after a directory share that directory. The raw archives are retained in the reproduction bundle; its runner reconstructs the exact union evidence. The paper and source map alone are insufficient to reproduce their observations. Direct source identities do not authenticate external claims or certify every transitive dependency.

The self-contained reproduction guide supplies a pinned Linux container and retained input archive. A fresh container with no host source mount or network reproduced all scores and accounting bytes; only the report timestamp changed. Restoration and analysis took 27.65 seconds on the author's machine, including 19.62 seconds in the unchanged analyzer. This is same-operator computational reproduction, not independent replication. The original A100 collection accumulated approximately 8.91 episode hours, excluding setup and gaps. The exact original Colab image and loaded CUDA/cuDNN inventory were not retained, so identical new model generation is not claimed.

The retained release checker, study/2026-1/release/verify.py, verifies prior release identities and selected arithmetic and component checks. It does not rerun every historical formal or conformance checker, reproduce the full training experiment or establish external replication. Missing early extraction records, some private conversation history and full old training trajectories remain recovery limits. Historical incident illustrations are not demonstrated prevention; no exploit reproduction or human-subject study was performed in the new local evaluation.

The full RR package reduced the specified structural exposures. Persistent defects often prevented completion, and coherent semantic errors remained unchecked. Its practical value depends on the consequences of both prevented violations and lost useful work.

Sources: analysis report, all scores, schedule accounting, unabridged verification, prior release verification.

Appendix. AI assistance and human direction

James Dimachkie set the research objective and directed its scope, interpretation and presentation through repeated exchanges with AI agents. His documented interventions included requesting evaluation of the full operational treatment, challenging missing or overstated claims, asking for comparison with prior work, and requiring a shorter, neutral account that preserved adverse findings. He selected the amended sample from the presented options and later closed further experimentation.

AI agents drafted and revised specifications, code, statistical arguments, analyses and manuscript text; they also operated the collection and verification tools. The following checks are documented in the retained artifacts.

Work	Documented verification
Treatment and host	Separate implementations, qualification cases, host review and trace checks
Statistical calculations	Exact rational computation and an internal reviewer's separate reconstruction of all six estimates and bounds
Task scoring	An independently authored deterministic scorer, development checks and sensitivity fixtures
Collected evidence	Checkpoint hashes, manifests, overlap checks, source identities and assignment accounting
Manuscript	Comparisons with source results, numerical checks, internal meaning review and rendered-page inspection

These were agent or programmatic checks within one operator's project. The record does not establish that James personally audited every formula, implementation or raw observation. It also does not support an exhaustive item-by-item classification of AI output as human-verified or accepted without review. No such classification is claimed. The human role described here concerns documented direction and decisions; an external human technical review has not yet been obtained.

Sources: qualification record, analysis report, unabridged verification, reproduction guide.