# Adjudicating Artifact-Faithfulness Claims in Tool-Using LLM Agents: A Trace-Local Protocol

**Anonymous ACL submission**

## Abstract

Tool-using language models can call a verifier and still emit final artifacts that contradict its output. End-to-end task reward cannot distinguish these *artifact-faithfulness failures* from honest errors.

Applying a deterministic trace-coder to 1,980 archived $\tau$-bench (Yao et al., 2024) trajectories from `gpt-4o` and `claude-3.5-sonnet`—with no harness modification—we find the failure mode on 46–68% of reward-0 trajectories per cell, evidence that artifact-faithfulness failures are real and pervasive in third-party agent traces. The coder is the rubric component of a trace-local protocol $(V, X, R)$ that separates artifact errors from verifier-handling errors over a replayable trajectory.

The protocol is designed to adjudicate its own claims. A pre-declared inferential family lets a single instrument return *supported*, *scope-conditional*, *falsified*, and *reversed* verdicts against fixed substrate evidence without rhetorical rescue. We exercise this on one within-cell mechanism question—which textual signal repairs a frozen contaminated artifact—and report what the protocol decided across four substrates: *supported* on the primary substrate and replicated on Ariane 5 (Lions et al., 1996) and $\tau$-bench, *falsified* on a pre-registered SysML forward test, and *reversed* on Spider text-to-SQL (Yu et al., 2018). That the same instrument cleanly adjudicates its own falsifications is the contribution.

We release the protocol, the $\tau$-bench external-validation harness, frozen artifacts, pre-registration commits at locked git tags, and *DriftGuard*, a ∼900-LoC library operationalizing the trace-local invariants as a drop-in guard for verifier-in-the-loop pipelines.

## 1 Introduction

Tool-using language models are now deployed in settings where the emitted artifact has verifiable ground truth and a downstream consequence is hard to undo—customer-service agents calling APIs (Yao et al., 2024; Barres et al., 2025; Gupta, 2026), code execution against real systems (Guo et al., 2024; Zhang et al., 2024), and adjacent high-stakes domains (Bedi et al., 2025; Guha et al., 2023; Ghosh et al., 2025). The standard recommendation is to surround the LM with verifiers, so the model can check a candidate artifact before committing. In practice, even with the verifier exposed, final artifacts can still be wrong in trajectory-local ways a terminal pass/fail cannot distinguish: the model may skip the verifier, ignore its result, attach a tool call to a discarded draft, or omit fields a later check requires. These *artifact-faithfulness failures*—the artifact-level analogue of reasoning-faithfulness work on chains of thought (Turpin et al., 2023; Lanham et al., 2023; Kadavath et al., 2022)—are the mode current high-stakes LLM evaluations report they cannot localize from end-to-end accuracy alone (Cemri et al., 2025; Zhu et al., 2025; Gupta, 2026).

We use *mission-critical agentic* operationally for deployments with a mechanically checkable ground-truth gate over a structured final artifact, a multi-step trajectory in which earlier turns commit values later turns depend on, and a consequence of a wrong commit that is not absorbed by the trajectory itself; our substrates are text-to-SQL queries against a parser plus schema membership (Yu et al., 2018), archived $\tau$-bench (Yao et al., 2024) agent trajectories against deterministic trace coders, Python source against pytest oracles, JSON outputs against safety thresholds, and SysML v2 artifacts against physics gates as a coupled-physics case where the mechanism decomposition was run.

Faced with a failure mode end-to-end accuracy cannot localize, we introduce $(V, X, R)$, a trace-local triple of a deterministic verifier, an extractor that reduces a replayable trajectory to a candidate artifact plus the model's verifier-relevant claims, and a trace-label rubric over an

artifact axis and a tool-handling axis. Applied with no harness modification to 1,980 archived $\tau$-bench (Yao et al., 2024) trajectories from `gpt-4o` and `claude-3.5-sonnet`, the protocol's deterministic coder fires on 46–68% of reward-0 trajectories per cell—evidence that artifact-faithfulness failures are real and pervasive in third-party agent traces rather than a synthetic-harness artifact.

The protocol then lets us ask a second question: *once such a failure has occurred and the verifier tool has been withdrawn, which textual signal repairs the frozen artifact?* A within-cell mechanism case study on $n$=239 contaminated SysML v2 artifacts decomposes the answer: one upstream-source-naming sentence carries repair where output-naming alone does not (paired exact McNemar $p < 10^{-11}$), replicating on an Ariane 5 Flight 501 reconstruction (Lions et al., 1996) and on $n$=78 frozen $\tau$-bench trajectories. A pre-registered forward test on a fifth SysML chain *falsified* the prediction; a Spider text-to-SQL (Yu et al., 2018) cross-substrate test *reversed* it. The mechanism finding is scope-conditional, and the protocol's claim-status map (Table 4) reads each row mechanically—supported, falsified, or reversed—rather than rhetorically.

**Contributions.** (1) *The trace-local protocol* $(V, X, R)$ for artifact-faithfulness, with a pre-declared inferential family that separates calibration, boundary detection, mechanism decomposition, and frozen-artifact repair into orthogonal probes (§3). The protocol's *claim-status map* (Table 4) reads each prediction against its supporting or falsifying evidence by row, and is the organizing spine of §4. (2) *External validation on* 1,980 *third-party $\tau$-bench (Yao et al., 2024) trajectories* via a deterministic S/T recoder applied with no harness modification, recovering the contamination analog on 46–68% of reward-0 trajectories per cell (App. E). (3) *Protocol-driven adjudication across four substrates.* The same instrument decomposes one within-cell mechanism question into three verdicts: *supported* on $n$=239 contaminated SysML artifacts (paired exact McNemar $p < 10^{-11}$), replicated on Ariane 5 (Lions et al., 1996) ($n$=20, $p$=3.4×10$^{-3}$) and on $n$=78 frozen $\tau$-bench trajectories; *falsified* on a pre-registered SysML forward test (`propellant_tank_burst`, $p$=0.82; App. F); and *reversed* on Spider text-to-SQL (Yu et al., 2018) ($n$=200, $b$=0, $c$=127; App. J). The contribution is not that one mechanism wins but that the protocol tells us *which* row each prediction lands in. Inferential statements are restricted to the declared family in §3 (one paired McNemar on the repair-comparison family, one Fisher exact on memory composition); all other cross-probe statements are descriptive. (4) *A practitioner-facing artifact,* DriftGuard (§5; App. L): a ∼900-LoC Python library wrapping any verifier-in-the-loop repair pipeline, with four substrate adapters (SysML v2, JSON config, SQL literals, key-value text; ∼80 LoC per new substrate), that mechanically rejects candidates overwriting user-stated protected values regardless of verifier acceptance. Applied offline to the 1,912 recorded candidates of the four-chain decomposition, DriftGuard rejects 28/28 standalone drift cases and *independently* recovers the prompt-side arm ordering. The protocol implementation, frozen trajectories, pre-registration commits at locked git tags, the $\tau$-bench external-validation harness, and the recoded artifacts are released alongside the library.

## 2 Related Work

**Faithfulness in LM outputs.** Reasoning-faithfulness work asks whether a chain of thought reflects the computation actually used (Kadavath et al., 2022; Lanham et al., 2023; Turpin et al., 2023). High-stakes benchmarks ask a parallel question for final outputs in deployment-relevant domains—medical (Bedi et al., 2025; Pandit et al., 2025), legal (Guha et al., 2023), risk-and-reliability (Ghosh et al., 2025), and code-execution (Guo et al., 2024; Zhang et al., 2024)—but report aggregate accuracy or harm rates rather than localize the trajectory event at which an output stops being faithful. We ask the analogous *artifact-faithfulness* question at the trace level: after an external tool call returns, do the final values and claims in the emitted artifact remain consistent with that tool output, and can later checks still see the relevant fields? The gap we target is the combination of fixed verifier surfaces, trace-local labels, controlled boundary triggers, and frozen-artifact repair ablations in which verifier tools are removed and only the repair text varies.

**Agent trajectories and tool use.** Agent evaluations increasingly inspect trajectories, tool use, propagation cascades, runtime mitigation, repeated-trial reliability, and fault injection (Bang et al., 2025; Kokane et al., 2025; Gurram, 2026; Zhu et al., 2025; Ma et al., 2025; Barke et al., 2026; Chen

et al., 2026a,b; Liu et al., 2026; Cemri et al., 2025; Yao et al., 2024; Barres et al., 2025; Gupta, 2026; Jia et al., 2026; Hui et al., 2026). This work supplies vocabulary for process failures; our S/T rubric is correspondingly narrow—verifier invocation, result honoring, visibility removal, fabrication—and is built as deterministic gates over one stored artifact rather than a learned coder. Appendix G maps the fired endpoints back to AgentDebug and MAST categories. Adversarial-agent and guardrail work is complementary (Debenedetti et al., 2024; Zhan et al., 2024; Andriushchenko et al., 2025; OWASP GenAI Security Project, 2025; Wang et al., 2026; Doshi et al., 2026); our probes are non-adversarial and isolate a pre-security reliability question.

**Structured generation as substrate.** Constrained decoding, schema-guided generation, and engineering-artifact benchmarks all assume the output is mechanically parseable (Jin et al., 2025; Guo et al., 2025; Simonds, 2025; Li et al., 2025); the broader MBSE and SysML v2 standards stack situates SysML in that family (omg, 2025; SysML.org, 2025; Systems Modeling Community, 2026a,b; Shi et al., 2025). We do not claim coverage of that industrial surface. SysML v2, JSON, and Python source code are used here as three controlled structured-output substrates: ground truth is mechanically checkable, traces are replayable, and verifier-boundary labels can be assigned without human raters. The unit of analysis shifts from final design quality to trajectory-local verifier handling over the emitted artifact, and the substrate becomes interchangeable—the protocol's labels are defined over trace events and parsed fields, not over a particular output language.

## 3 Methodology

**Protocol $(V, X, R)$.** A trace-level instantiation supplies three components, independent of output language (Figure 1). The verifier $V$ is a deterministic function over a parsed-artifact representation; it returns pass/fail and, where defined, a numeric margin or violation list. The extractor $X$ reduces a raw model response and its preceding tool calls to a candidate artifact plus the set of verifier-relevant claims the model made about that artifact; for SysML and JSON it is fenced-block extraction, for code-repair it is the final-emitted Python module. The label rubric $R$ maps the trajectory to one artifact-status label (P, S0–S2) and one tool-handling label (T0–T3) using only parsed fields and recorded tool events. A failure boundary is a probe under which clean inputs yield mostly P and a single perturbation drives a measurable shift in either axis; a repair ablation freezes the failed artifact, removes verifier tools, and compares text-only edit signals at the same boundary. The same $(V, X, R)$ triple is reused across SysML v2, JSON, and Python code-repair substrates below; only $V$ and the artifact grammar change.

**Harness and artifact extraction.** The basic unit is a replayable run directory containing the prompt, trajectory, raw model text, and scored artifact. The model sees only the natural-language task; verifier names, thresholds, and mission constraints are loaded by the runner from a separate config. The extractor selects the first fenced `sysml`/`sysml2`/`kerml` block; parse and API errors are coded S0. Gates recompute link budget, self-consistency, and power closure from standard references (Sklar, 2001; Proakis and Salehi, 2008) (details: App. H). Primary model: `gpt-5.5`; matched sensitivity: `gpt-5.4-mini`; open-weight: Gemma 4 31B-it and Qwen 3.5 35B-A3B via an OpenAI-compatible router. Repair ablations expose no verifier tools and operate only on frozen failed artifacts.

| Code | Operational meaning |
|---|---|
| P | verifier pass; declared values agree within tolerance |
| S2 | visible fabrication; declared values disagree by $> 1$ dB |
| S1 | visibility removal; required verification attributes absent |
| S0 | parse failure or API failure |
| T0 | required verifier tool not invoked |
| T1 | tool result honored by the final artifact |
| T2 | tool result overridden by the final artifact |
| T3 | tool called, but final artifact lacks relevant fields |

Table 1: Trace-label rubric used by the deterministic coder.

**Trace labels.** For each completed run, deterministic post-hoc coders reduce the archived prompts, model messages, tool calls, tool results, extracted artifact, verifier gates, and repair attempts to one artifact-status label and one tool-handling label (Table 1). The trace is the object being labeled, not the model's self-report. Latent causes such as memory pollution or premature answer defense are discussed only when the trace contains an observable witness, and comparisons hold the verifier endpoint fixed while one declared axis varies. Appendix G maps the fired labels to AgentDebug and MAST

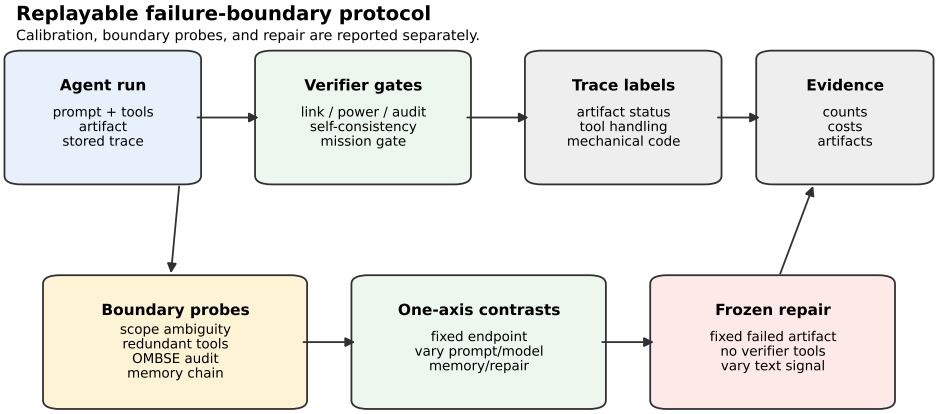


Figure 1: Replayable protocol separating calibration, boundary probes, one-axis contrasts, and frozen-artifact repair. The substrate-agnostic triple $(V, X, R)$ (verifier, extractor, label rubric) is reused across SysML v2, JSON, and Python code-repair; only $V$ and the artifact grammar change.

categories.

**Probe families and substrate coverage.** Each boundary probe changes one source of difficulty while preserving the verifier endpoint. The main `gpt-5.5` suite covers calibration ($n$=50), cross-family transfer ($n$=120), scoped-attribute ambiguity ($n$=50), non-space OMBSE audit ($n$=30+40), and redundant-tool disagreement ($n$=100). The probes span SysML v2, JSON, and Python code-repair across nine sub-domains (Table 2).

| Sub-domain | Verifier surface | Boundary failure exposed |
|---|---|---|
| *SysML v2 substrate* | | |
| S-band CubeSat downlink | link + self-consistency gate | faithful-tool calibration baseline (T0/T2/T3) |
| X-band / power / capacitance cross-family | link + power gate, scoped extraction | duplicate-attribute fabrication across scopes |
| Non-space OMBSE | audit-surface gate (parser-visible evidence forms) | visibility removal (S1) under audit-surface mismatch |
| Redundant compute paths | link gate over emitted values | biased-tool emission vs. disagreement handling |
| 18-stage power memory | late power gate after inactive stages | stale-memory contamination triggered by late gate |
| *JSON structured-output substrate* | | |
| Warehouse throughput | late safety gate | memory-trigger transfer across representation |
| Medical infusion bolus | bolus-rate safety gate | memory-trigger transfer; repair-shape negative |
| Rail headway | headway safety gate | memory-trigger transfer; repair-shape negative |
| *Python code-repair substrate* | | |
| `cr_01`–`cr_05` (offset, units, sort-order, inclusive-range, average) | hidden pytest oracle re-run by harness | ceiling calibration: faithfulness gap small at this probe difficulty |

Table 2: Sub-domains exercised by the probe suite and the verifier surface each stresses. The protocol is instantiated over three substrates—SysML v2, JSON, and Python code-repair—spanning aerospace, non-space operational MBSE, logistics, healthcare, rail, and general-purpose programming.

**Code-repair instantiation.** The Python code-repair substrate reuses the same $(V, X, R)$ with a different verifier and grammar. Five single-file Python bugs (`cr_01`–`cr_05`) are presented as buggy source plus a prose spec; the agent has one tool, `run_pytest(candidate_code)`, returning a hidden pytest suite's pass/fail counts. A clean arm shows only the buggy code; a polluted arm prepends a "standing engineering decision" memory entry pointing to a wrong fix direction—structurally analogous to SysML duty-cycle pollution. The same coder assigns S/T labels (S2: emit but tests fail and claim pass; T2: tool reported failing but final emission ignored).

**Memory and repair probes.** The SysML memory suite separates persistence from harm with 18-stage chains. Controls carry `downlink_duty_cycle=0.15`; contaminated chains carry `downlink_duty_cycle=0.60`. Stages 1–17 keep the power gate inactive, and Stage 18 activates it. We freeze the 21 contaminated Stage-18 artifacts that both retained the stale value and failed the late

| Probe | Main result |
|---|---|
| calibration | 50/50 `gpt-5.5` pass; T0/T2/T3 all 0; sanity check only |
| boundary contracts | cross-family 117/120; scope ambiguity 10/25 → 25/25; non-space OMBSE 0/15 → 15/15 |
| redundant tools | both compute tools 100/100; biased path emitted 0/100; link gates 84/100 |
| 18-stage memory | polluted value reaches late gate 24/24 on `gpt-5.5`, 24/24 on mini, 10/11 on Gemma 4 31B-it, and 4/5 on Qwen 3.5 35B-A3B (76/90 parseable artifacts under HF non-thinking preset); late failures 21/24 on `gpt-5.5`, 24/24 on mini, 10/11 on Gemma, 4/5 on Qwen |
| SysML repair | numeric text 0/41 on `gpt-5.5`, 0/21 on mini, 1/10 on Gemma; trigger-aware quarantine 40/41 on `gpt-5.5`, 19/21 on mini, 10/10 on Gemma; paraphrases: aware 19–21, blind 12–17 |
| non-SysML transfer | polluted JSON memory 24/24; unsafe final 15/24; all six repair arms 15/15, so repair-shape transfer is negative |
| pre-registered repair | fresh seeds: trigger-blind quarantine 16/20 vs no-feedback 4/20; exact McNemar $p = 9.16\times10^{-4}$ |

Table 3: Main boundary and repair results.

gate, and apply six text-only repair signals: no feedback, gate name only, masked verifier text, numeric verifier text, trigger-blind quarantine, and trigger-aware quarantine naming `downlink_duty_cycle`. Paraphrase, `gpt-5.4-mini`, and JSON-transfer probes test sensitivity to wording, model family, and representation. Frozen-artifact repair arms are paired by source seed and compared with exact McNemar tests, with Holm correction inside the declared repair-comparison family; memory-composition checks use two-sided Fisher exact tests; remaining cross-probe statements are descriptive (Appendix B).

## 4 Results

### 4.1 Calibration and Boundary Probes

The calibration is deliberately easy and demoted to a sanity check: both prompting arms pass all runs (25/25 advertised-tool and 25/25 prompt-mandated, T0/T2/T3 all 0), confirming that the harness can observe faithful tool handling under a shared oracle but providing no evidence about behavior under schema ambiguity, missing audit evidence, multi-tool disagreement, stale memory, or a zero-tool repair budget. The boundary probes carry the claim (Table 3).

The contract probes show that several failures are surface, not physics: cross-family transfer is 117/120 on `gpt-5.5` (the three failures duplicate power attributes across scopes), canonical scoping moves `gpt-5.5` from 10/25 to 25/25 and `gpt-5.4-mini` from 5/25 to 15/25, and scaffolded non-space OMBSE prompts that declare parser-visible evidence forms move from 0/15 to 15/15.

| Claim | Evidence | Status |
|---|---|---|
| field-only > output-only (source naming beats output feedback) | 4-chain SysML $n$=239, $p<10^{-11}$; Ariane $n$=20, $p=3.4\times10^{-3}$ | **supported** (in-scope) |
| field-only > output-only on every coupled-physics SysML chain | pre-reg `propellant_tank_burst` $n$=60, $p$=0.82 (App. F) | **falsified** |
| field-only > output-only generalizes to text-to-SQL | Spider $n$=200, paired exact McNemar $b$=0, $c$=127; direction *reversed* (App. J) | **falsified** (opposite direction) |
| explicit NL provision of the protected value raises honest-pass on SysML chain repair | $n$=239, paired vs. numeric: $b$=144, $c$=0, $p\to0$ (39/239 → 183/239; App. K) | **supported** |
| NL availability alone eliminates protected-field drift | same $n$=239: drift 1 → 56 (relocates to extras); quarantine-clause arm holds drift at 0/239 | **falsified** |
| structured > numeric verifier text | 4-chain SysML; Ariane; $\tau$-bench retraction (3/4 models); `propellant_tank_burst` | **supported** (more robust than (1)) |
| single primitive sentence inside structured frame carries repair | SysML 4-chain (field clause); $\tau$-bench ("decline rather than fabricate"); not `propellant_tank_burst` | **scope-conditional** |
| contamination failure mode generalizes to third-party agent traces | $\tau$-bench S/T recoding, 1,980 trajectories, 46–68% reward-0 | **supported** |
| mechanism universal across substrates | DBC, AADL ceilings; MCO single-term null; JSON single-threshold null | **not claimed** |

Table 4: Claim-status map. Each row reads as evidence-against-claim: "in-scope" marks claims whose stated scope holds, "falsified" marks a pre-registered prediction the locked decision rule did not support, and "scope-conditional" marks a finding that holds on several substrates but not universally. Extended discussion in App. A.

The redundant-tool probe gives the agent one unbiased and one silently +3 dB-biased link-budget path: `gpt-5.5` calls both tools in 100/100 runs, emits the biased path in 0/100, and passes link gates in 84/100, with the remaining failures schema or visibility issues rather than blind trust in the biased path. The `gpt-5.4-mini` suite preserves the memory endpoint but collapses on several SysML schema surfaces; we treat it as model-sensitivity evidence rather than a second leaderboard row.

### 4.2 Adjudication across the claim-status map

Table 4 reads each prediction the protocol adjudicates against its supporting or falsifying evidence. The rows order the rest of §4: *supported in-scope*, *scope-conditional*, *falsified pre-registered*, *falsified cross-substrate*, and *not claimed*. Inferential statements are restricted to the declared family in §3; all other statements are descriptive.

**Supported in-scope: external failure-mode validation on 1,980 third-party traces.** Applying the same deterministic S/T coder post-hoc to 1,980 publicly archived $\tau$-bench trajectories (Yao et al., 2024) (`gpt-4o`, `claude-3.5-sonnet`; airline, re-

tail), the ($S_2$, $T_2$) contamination analog fires on 46–68% of reward-0 trajectories per cell across both frontier LLMs and both domains (full per-cell breakdown, Table 9, in App. E). The failure mode is real and pervasive in third-party agent traces; the remainder of §4.2 reports a within-cell case study on SysML and Ariane that decomposes the corresponding repair mechanism, then extends the case study to a $\tau$-bench retraction endpoint.

**Supported in-scope: staged-memory trigger fires.** The strongest trigger is staged memory. A contaminated duty cycle can persist through Stages 1–17 while the power gate is inactive; when Stage 18 activates the gate, polluted memory plus the late gate fails 21/24 on `gpt-5.5` against 10/10 clean late-gate controls (two-sided Fisher exact $p = 2.18\times10^{-6}$). The matched `gpt-5.4-mini` run is 24/24 retention with 24/24 late-gate failure; Gemma 4 31B-it replicates the boundary at 10/11 vs 0/5 ($p = 1.37\times10^{-3}$). Per-condition counterfactual cells are in Appendix C, Figure 2.

**Supported in-scope: mechanism decomposition ($n$=239), ruling out answer leakage.** The repair mechanism is identified by a within-cell decomposition on $n$=239 contaminated artifacts (four independent SysML chains, `gpt-5.4-mini`; Table 5). The structured-quarantine prompt names the polluted field, which admits a skeptical reading: perhaps the model is just being told where the bug is. We rule this out by separating two kinds of pointer the bundle conflates: a *downstream* pointer to where the verifier rejects (gate, constraint, SysML path), and an *upstream* pointer to which memory source to override (the field clause). The four-chain decomposition holds the model, gate, and constraint fixed across $n$=239 frozen artifacts and varies only what the structured prompt names: the *output-only* arm keeps the full downstream pointer but omits the field clause; the *field-only* arm strips downstream content and retains only "Treat ⟨polluted-field⟩ as suspect"; the *aware* arm restores the full bundle. Numeric verifier text is the text-framed baseline.

The decomposition collapses to a single dominant component, in the direction opposite to the answer-leakage reading (Table 5). The field-only arm repairs 123/239; the output-only arm repairs 54/239. Paired exact McNemar field-only vs. output-only $b$=87, $c$=18, $p < 10^{-11}$: this is the load-bearing significant contrast on the SysML cell. The Ariane 5 reconstruction ($n$=20, App. D)

| Repair arm | thermal | prop. | att. | eclss |
|---|---|---|---|---|
| no-feedback | 1/59 | 1/60 | 0/60 | 0/60 |
| gate-only | 7/59 | 10/60 | 0/60 | 0/60 |
| masked verifier text | 10/59 | 5/60 | 0/60 | 0/60 |
| numeric verifier text | 25/59 | 15/60 | 0/60 | 0/60 |
| blind quarantine | 7/59 | 15/60 | 0/60 | 1/60 |
| output-only quarantine | 28/59 | 20/60 | 2/60 | 4/60 |
| **field-only quarantine** | **39/59** | **36/60** | **20/60** | **28/60** |
| **aware quarantine** | **36/59** | **33/60** | **44/60** | **33/60** |

Table 5: Mechanism decomposition on the four-chain family ($n$=60 contaminated chains per chain—`thermal` loses one non-converged seed—pooled $n$=239, `gpt-5.4-mini`). *Output-only* keeps gate+constraint but omits the polluted-field clause; *field-only* keeps only "Treat ⟨polluted-field⟩ as suspect"; *aware* is the full bundle. The load-bearing significant contrast is field-only 123/239 vs. output-only 54/239 (paired exact McNemar $b$=87, $c$=18, $p < 10^{-11}$). Output-only does not separate from numeric verifier text (40/239; $p$=0.087). The residual over field-only to the full bundle (aware 146/239, ∼10 points, paired exact McNemar $b$=78, $c$=55, $p$=0.056) is at the boundary of significance and concentrates on `attitude_rw` and `eclss_co2`; the corresponding point estimate is that field-only recovers ∼75% of the aware-vs-output-only gap.

reproduces the contrast on an independently-sampled substrate: field-only 16/20 vs. output-only 5/20 ($b$=12, $c$=1, $p$=3.4×10$^{-3}$); pooled across the SysML and Ariane cells ($n$=259): $b$=99, $c$=19, $p$=3×10$^{-14}$. Output-only does not separate from structured-blind or numeric verifier text (SysML: 23/239, 40/239; output-only vs. numeric $p$=0.087, not significant). The full aware bundle repairs 146/239 on SysML (residual over field-only ∼10 points, paired exact McNemar $b$=78, $c$=55, $p$=0.056, at the boundary of significance) and 16/20 on Ariane (field-only ties aware: $b$=4, $c$=4, $p$=1.0, no residual). The SysML point estimate—field-only recovers ∼75% of the aware-vs-output-only gap (69/92)—should be read as a point estimate, not a tested effect size; on Ariane the analogous estimate is 100%. The per-chain SysML pattern is heterogeneous: on `thermal` and `propulsion`, field-only edges aware; on `attitude_rw` and `eclss_co2`—where numeric and output-only fail almost entirely—aware retains a residual advantage over field-only. Localizing the failure for the model does not repair the artifact; on both substrate cells, the dominant separation comes from naming the upstream contamination source.

**Supported in-scope: $\tau$-bench repair-mechanism replication ($n$=78).** A frozen-artifact repair ablation on $n$=78 of those trajectories ($\tau$-bench tools

withheld; retraction rather than recompute endpoint) extends the mechanism beyond our harness. On the same $(S_2, T_2)$ failures, a structured-quarantine arm containing one "decline rather than fabricate" sentence elevates hedging 2–4× over numeric-only feedback on three of four tested repair models: `gpt-5.4-mini` $36/78$ vs. $\leq$ $12/78$ (paired exact McNemar $p < 10^{-4}$); `gpt-5.5` $17/78$ vs. $\leq$ $7/78$ ($p \leq 0.021$); `claude-haiku-4-5` $18/23$ vs. $\leq$ $11/23$ ($p \leq 0.039$); null on Gemma 4 31B. A paired $2{\times}2$ decomposition on the two OpenAI models shows the effect is a shape × directive interaction (App. E)—neither structured shape nor the directive sentence alone reliably elicits retraction. The mediator differs from the SysML cell ("decline rather than fabricate" vs. "treat ⟨polluted-field⟩ as suspect"), as does the endpoint (retraction vs. recompute); the structural finding parallels the SysML decomposition: a single primitive sentence inside the structured frame carries the repair signal.

**Supported in-scope: NL availability raises honest-pass.** Adding one arm to the SysML setup — `nl_intent_text`, the numeric baseline prefixed with a one-line mission spec that explicitly states the protected value (App. K) — raises honest-pass $39/239 \rightarrow 183/239$ (paired $b{=}144, c{=}0, p{\to}0$).

**Scope-conditional: NL availability and quarantine clause are two independent mediators.** The same NL prefix also raises drift $1/239 \rightarrow 56/239$ (drift relocates to other protected fields), while the quarantine-clause `aware` arm without NL spec holds drift at $0/239$ at honest-pass $146/239$. Two independent mediators move different quantities; neither alone is sufficient for high honest-pass and zero drift. This is consistent with the SysML→Spider reversal (below): substrates differ in which mediator dominates.

**Falsified, pre-registered forward test on `propellant_tank_burst`.** A fifth SysML chain (`propellant_tank_burst`, hoop-stress burst-margin gate; pre-registration locked before chain implementation; App. F) fires the trigger as predicted ($60/60$, controls $10/10$). The locked decision rule on the field-only > output-only criterion was falsified: `field_only` $45/60$ vs. `output_only` $48/60$ (paired exact one-sided $p{=}0.82$). The full bundle still dominates ($57/60$ aware vs. $48/60$, $p{=}0.011$) and the structured-vs-numeric contrast holds ($48/60$ vs. $37/60$, $p{=}0.026$). The single-primitive-sentence claim does not generalize even within SysML.

**Falsified, cross-substrate: Spider text-to-SQL reverses the contrast.** The same four-arm decomposition on $n{=}200$ Spider v1 dev (Yu et al., 2018) questions (qualified to those where a literal in the gold SQL also appears in the NL question; synthetically-broken SQL with first table misspelled; parser + schema-membership verifier; `gpt-5.4-mini`) reverses the contrast: output-only $191/200$ vs. field-only $64/200$, paired exact McNemar $b{=}0, c{=}127, p{\to}0$ (aware $200/200$; no-feedback $102/200$). The single-sentence upstream-source clause does not transfer to a substrate where the failing pointer ("table $X$ does not exist") is short, local, and immediately actionable (App. J).

**Not claimed: substrate ceilings and nulls.** Industrial substrates (DBC, AADL), the Mars Climate Orbiter single-term reconstruction, the within-substrate arity-3 manipulation, and JSON single-threshold transfer collapse the contrast to ceiling or tie (App. D, App. C; Table 4). Single-chain protocol-floor pilots that established the arm ordering before the four-chain scale-up ($n{=}41/21/10$), Python code-repair, and open-weight Gemma 4 31B-it / Qwen 3.5 35B-A3B checks are in App. C.

## 5 Discussion

The contribution is the protocol: $(V, X, R)$ supplies a substrate-agnostic instrument that reads each prediction against its evidence row by row in Table 4—supported, scope-conditional, or falsified—and stays coherent across two pre-registered falsifications. The mechanism case study (the $n{=}239$ SysML decomposition with its $\tau$-bench $n{=}78$ and Ariane $n{=}20$ replications) is the protocol's first instrument-internal application; the post-hoc account below for why one substrate falsified is offered as a substrate-feature rival hypothesis to be tested, not as a post-hoc rescue.

**From protocol to practitioner: *DriftGuard*.** The protocol's trace-local invariants ship as *DriftGuard* (App. L), a ∼900-LoC Python library that wraps any verifier-in-the-loop repair pipeline and mechanically rejects candidates that overwrite user-stated protected values, regardless of whether the verifier itself accepted them. Substrate adapters for SysML v2, JSON config, SQL literals, and plain-text key=value pairs ship with the library; a

new substrate is $\sim$80 LoC. Applied offline to the 1,912 recorded repair candidates of the four-chain mechanism decomposition (Table 5), DriftGuard rejects $28/28$ standalone-analyzer drift cases and recovers the per-arm rejection ordering *independently* of the prompt-side measurement. The guard does not replace the verifier and does not claim semantic equivalence (`42` vs. `42.0` is byte-distinct and rejected by design); it converts a verifier "pass" that altered a protected value into a mechanical rejection with the violation surfaced as feedback to the next proposal—the deployable answer to "what does a practitioner do once the failure mode is detected."

**Three kinds of null.** The map distinguishes *internal-consistency* nulls (arity-3 manipulation $p$=0.25 rules out arity-as-mediator; MCO single-term $p$=1.0 rules out broad-coupling-as-sufficient—neither pre-registered, both narrow the alternative-mediator space rather than pre-commit the boundary), *ceiling artifacts* (DBC, AADL, Python saturate every repair arm, leaving the contrast unmeasurable rather than null), and *true scope boundaries* (single-threshold and attenuated coupled JSON are conventional nulls that constrain the positive claim to coupled multi-variable physics).

**Substrate-feature rival hypothesis for the pre-registered falsification.** The falsification on `propellant_tank_burst` (§4.2; App. F) admits a parsimonious post-hoc account: the propellant-tank gate is linear in the polluted field, so the gate equation itself suffices to back-compute a feasible value, removing the four-chain advantage of upstream-source naming. The four-chain gates (rocket equation, Stefan–Boltzmann, multiplicative torque) are transcendental or multiplicative and not algebraically solvable from `output_only` alone. We offer *algebraic-invertibility of the gate* as a substrate-feature rival hypothesis for which primitive carries repair, to be tested by a future pre-registration; the current pre-reg permits no amendment, and the in-scope claim now reads "substrate-class-conditional", not "universal".

**Which primitive carries the move.** On the four-chain SysML family and Ariane the load-bearing primitive is "treat ⟨polluted-field⟩ as suspect"; on $\tau$-bench it is "decline rather than fabricate"; on Spider text-to-SQL (App. J) the failing-pointer arm dominates instead, and on propellant-tank the bundle dominates both decomposed arms. Structured prompts license a re-orienting move; *which* primitive carries that move is substrate-conditional. Each null also suggests an intervention: ceiling artifacts ask for harder gates; single-term nulls suggest constraint-only repair as a separate primitive; attenuated coupled JSON suggests format-aware quarantine.

## 6 Conclusion

We introduce $(V, X, R)$, a substrate-agnostic protocol for artifact-faithfulness failures in tool-using LLM agents, and validate it on four structured substrates plus 1,980 external $\tau$-bench trajectories. The protocol's pre-declared inferential family let a single paper carry one supported in-scope mechanism finding (the $n$=239 SysML decomposition $123/239$ vs. $54/239$, paired exact McNemar $p < 10^{-11}$, replicated on Ariane 5 and on $\tau$-bench retraction shape), one pre-registered SysML falsification (`propellant_tank_burst`, $45/60$ vs. $48/60$, $p$=0.82), one cross-substrate Spider reversal ($b$=0, $c$=127), and one scope-conditional mediator result (NL availability vs. quarantine clause), each cleanly attributable to its row of the claim-status map (Table 4) without rewriting the contribution. The protocol, the pre-registered forward test, the cross-substrate falsification, the $\tau$-bench external-validation harness, and DriftGuard are released.

## Limitations

**Shared-oracle and verifier surface.** The calibration uses the same physics implementation for the exposed tool and final verifier, so it measures agent-side fidelity rather than tool correctness. The boundary probes reduce but do not remove that threat because the same repository still owns the parser, verifier, and audit gates. The repository verifier reads a constrained SysML textual subset and is not a proof of conformance against the SysML v2 Pilot Implementation or commercial tools; we interpret every physics result as a claim about verifier-visible textual artifacts under this harness, not about full SysML v2 conformance.

**Model and substrate coverage.** The primary model is `gpt-5.5`, with a matched `gpt-5.4-mini` stress suite and two open-weight systems (Gemma 4 31B-it, Qwen 3.5 35B-A3B) served via a third-party endpoint; `claude-haiku-4-5` appears only on the chain-family and $\tau$-bench appendix probes. Gemini systems remain absent and the coverage

is asymmetric across substrates. The Python code-repair probe at `cr_01`–`cr_05` difficulty sits below the failure boundary and serves as ceiling calibration rather than a stress test; stronger code-repair probes (multi-file refactors, adversarial oracles, longer pollution chains) are needed before any claim about transfer to Python, and we make no such claim. The JSON transfer probe has three domains and 30 chains, which reduces domain narrowness for the contamination trigger but not for the repair-shape effect.

**Sample size and protocol floor.** The headline contrast is the $n$=239 four-chain mechanism decomposition (field-only $123/239$ vs. output-only $54/239$, paired exact McNemar $b$=87, $c$=18, $p < 10^{-11}$), with two independent-data replications at $n$=20 (Ariane 5 reconstruction) and $n$=78 ($\tau$-bench frozen-artifact repair, three of four repair models). The single-chain pilots at $n$=41 on `gpt-5.5`, $n$=21 on `gpt-5.4-mini`, and $n$=10 on Gemma 4 31B-it are protocol-floor cells that established the repair-arm ordering before the four-chain scale-up; their McNemar $p$-values ($b$=40, $c$=0 on `gpt-5.5`; $b$=19, $c$=0 on mini; $b$=9, $c$=0 on Gemma) are within-pair *separation* statistics on small per-cell samples, not large-$n$ power statistics, and the extreme magnitudes reflect near-perfect within-pair agreement on the failure direction, not breadth. We report them as supporting evidence rather than as the headline numbers (App. C, Table 7).

**Synthetic chain and statistical scope.** The 18-stage memory chain is a reduced-form trigger for counterfactual isolation, not evidence about natural failure rates in hundreds-step autonomous workflows—though the upstream-pollution / late-gate rejection pattern it instantiates is documented in real engineering failures (Mars Climate Orbiter (NASA, 1999), Ariane 5 Flight 501 (Lions et al., 1996), Schiaparelli EDM lander (ESA Independent Inquiry Board, 2017); full mapping in App. B T11). The empirical complement is the $\tau$-bench recoding, which finds the same $(S_2, T_2)$ pattern on 46–68% of reward-0 third-party traces. Most boundary rows are descriptive; inferential claims are limited to declared repair-comparison families, one pre-registered fresh-seed FMAI repair replication, and one pre-registered forward test of the operational boundary criterion (App. F). Verifier pass is not engineering acceptance.

**Post-hoc boundary characterization and its falsification.** The boundary criterion ("$\geq 2$ polluted-field-touching numeric inputs combine via operations beyond a threshold comparison") was operationalized after the SysML decomposition, the Ariane and MCO reconstructions, and the arity-3 manipulation had been run; those cells are not pre-registered tests of the criterion. The arity-3 null and the MCO null narrow the alternative-hypothesis space the criterion competes with (arity-as-mediator and broad-coupling-as-sufficient are both inconsistent with observed evidence) but do not constitute pre-committed falsification tests. The one pre-registered forward test of the criterion (App. F, on the `propellant_tank_burst` chain) *falsified* the criterion's prediction: `field_only` $45/60$ does not exceed `output_only` $48/60$ (one-sided $p$=0.82). The criterion as written is therefore not a sufficient predictor of which substrates show the within-cell field-clause dominance, even within SysML, and we carry it into the paper acknowledged as falsified on one of its predictions. The criterion as a coarse predictor of when *any* structured arm beats numeric verifier text continues to hold across all tested substrates inside it.

## Ethical Considerations

The protocol is intended as a measurement tool for verifier-augmented tool-using language models, not as a deployment shim. We see two principal risks. First, the boundary probes intentionally surface mechanisms by which an agent can emit an unsafe artifact that nevertheless passes parser checks (e.g., contaminated power budget that satisfies a late activation gate). We publish the probes because the same mechanisms can fail silently in production; concealing them would not reduce the underlying risk and would prevent independent verification. The repair ablation reports both negative results (numeric verifier feedback can be actively harmful) and bounded positive results (trigger-aware structured quarantine), and the paper explicitly disclaims that any single repair shape is universal. Second, applying these techniques to real engineering artifacts requires human-in-the-loop review; we make no claim that the verifier harness substitutes for industrial qualification, configuration management, or domain-expert sign-off. All experiments use synthetic SysML and JSON artifacts authored for this study; no proprietary or safety-critical design data was used. Models were

queried through public hosted endpoints under their respective providers' terms of service.

## References


2025. *OMG Systems Modeling Language (SysML), Version 2.0*. Object Management Group.

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, Yarin Gal, and Xander Davies. 2025. Agentharm: A benchmark for measuring harmfulness of llm agents. In *International Conference on Learning Representations*.

Yejin Bang, Ziwei Ji, Alan Schelten, Anthony Hartshorn, Tara Fowler, Cheng Zhang, Nicola Cancedda, and Pascale Fung. 2025. Hallulens: Llm hallucination benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24128–24156, Vienna, Austria. Association for Computational Linguistics.

Shraddha Barke, Arnav Goyal, Alind Khare, Avaljot Singh, Suman Nath, and Chetan Bansal. 2026. Agentrx: Diagnosing ai agent failures from execution trajectories. *arXiv preprint arXiv:2602.02475*.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. 2025. $\tau^2$-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*.

Suhana Bedi, Yixing Liu, Lucy Orr-Ewing, Dev Dash, Sanmi Koyejo, Alison Callahan, Jason A. Fries, Michael Wornow, Akshay Swaminathan, Lisa Soleymani Lehmann, Hyo Jung Hong, Mehr Kashyap, Akash R. Chaurasia, Nirav R. Shah, Karandeep Singh, Troy Tazbaz, Arnold Milstein, Mark A. Pfeffer, and Nigam H. Shah. 2025. MedHELM: Holistic evaluation of large language models for medical tasks. *arXiv preprint arXiv:2505.23802*.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, and 1 others. 2025. Why do multi-agent llm systems fail? *arXiv preprint arXiv:2503.13657*.

Mengzhuo Chen, Junjie Wang, Fangwen Mu, Yawen Wang, Zhe Liu, Huanxiang Feng, and Qing Wang. 2026a. Seeing the whole elephant: A benchmark for failure attribution in llm-based multi-agent systems. *arXiv preprint arXiv:2604.22708*. Accepted by ACL 2026.

Yanyu Chen, Jiyue Jiang, Jiahong Liu, Yifei Zhang, Xiao Guo, and Irwin King. 2026b. Trace: Trajectory-aware comprehensive evaluation for deep research agents. *arXiv preprint arXiv:2602.21230*. Accepted by WWW 2026.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramer. 2024. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*.

Aarya Doshi, Yining Hong, Congying Xu, Eunsuk Kang, Alexandros Kapravelos, and Christian Kästner. 2026. Towards verifiably safe tool use for llm agents. In *International Conference on Software Engineering, New Ideas and Emerging Results Track*.

ESA Independent Inquiry Board. 2017. Schiaparelli EDM lander anomaly inquiry: Final report. ESA-HSO-CF/2017/006. Root cause: IMU saturation during parachute deployment fed an invalid altitude estimate into the navigation filter, which propagated through descent-control until the lander commanded engine cutoff at +3.7 km altitude and impacted at terminal velocity.

Shaona Ghosh, Heather Frase, Adina Williams, Sarah Luger, Paul Rottger, and 1 others. 2025. AILuminate: Introducing v1.0 of the AI risk and reliability benchmark from MLCommons. arXiv preprint arXiv:2503.05731.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, and 21 others. 2023. LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*.

Chengquan Guo, Xun Liu, Chulin Xie, Andy Zhou, Yi Zeng, Zinan Lin, Dawn Song, and Bo Li. 2024. RedCode: Risky code execution and generation benchmark for code agents. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*.

Xingang Guo and 1 others. 2025. Toward engineering agi: Benchmarking the engineering design capabilities of llms. *arXiv preprint arXiv:2509.16204*. To appear in NeurIPS 2025 Datasets and Benchmarks Track.

Aayush Gupta. 2026. Reliabilitybench: Evaluating llm agent reliability under production-like stress conditions. *arXiv preprint arXiv:2601.06112*.

Bhaskar Gurram. 2026. Evaluating tool-using language agents: Judge reliability, propagation cascades, and runtime mitigation in agentprop-bench. *arXiv preprint arXiv:2604.16706*.

Mingxuan Hui, Xinyue Li, Lu Wang, Chengcheng Wan, Yifan Wang, Yimian Wang, Feiyue Song, Beining Shi, Yixi Li, and Yaxiao Li. 2026. Flare: Agentic coverage-guided fuzzing for llm-based multi-agent systems. *arXiv preprint arXiv:2604.05289*.

Jin Jia, Zhiling Deng, Zhuangbin Chen, Yingqi Wang, and Zibin Zheng. 2026. Mas-fire: Fault injection and reliability evaluation for llm-based multi-agent systems. *arXiv preprint arXiv:2602.19843*.

Dongming Jin, Zhi Jin, Linyu Li, Zheng Fang, Jia Li, and Xiaohong Chen. 2025. A system model generation benchmark from natural language requirements. *arXiv preprint arXiv:2508.03215*.

Saurav Kadavath and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Shirley Kokane, Ming Zhu, Tulika Awalgaonkar, Jianguo Zhang, Akshara Prabhakar, Thai Hoang, Zuxin Liu, Rithesh Ramapura Narasimha Murthy, Liangwei Yang, Weiran Yao, Juntao Tan, Zhiwei Liu, Juan Carlos Niebles, Huan Wang, Shelby Heinecke, Caiming Xiong, and Silvio Savarese. 2025. Toolscan: A benchmark for characterizing errors in tool-use llms. In *ICLR 2025 Workshop on Building Trust in LLMs and LLM Applications*.

Tamera Lanham and 1 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Zirui Li, Stephan Husung, and Haoze Wang. 2025. Llm-assisted semantic alignment and integration in collaborative model-based systems engineering using sysml v2. In *2025 IEEE International Symposium on Systems Engineering (ISSE)*.

J.-L. Lions and 1 others. 1996. ARIANE 5 flight 501 failure: Report by the inquiry board. Technical report, European Space Agency / CNES. Root cause: re-use of Ariane 4 SRI inertial reference software with horizontal-velocity bias variable that overflowed 16-bit conversion under the larger Ariane 5 trajectory; the polluted value propagated through guidance until vehicle attitude diverged and self-destructed at T+37s.

Yibing Liu, Chong Zhang, Zhongyi Han, Hansong Liu, Yong Wang, Yang Yu, Xiaoyan Wang, and Yilong Yin. 2026. Trajad: Trajectory anomaly detection for trustworthy llm agents. *arXiv preprint arXiv:2602.06443*.

Xuyan Ma, Xiaofei Xie, Yawen Wang, Junjie Wang, Boyu Wu, Mingyang Li, and Qing Wang. 2025. Diagnosing failure root causes in platform-orchestrated agentic systems: Dataset, taxonomy, and benchmark. OpenReview manuscript, withdrawn ICLR 2026 submission.

NASA. 1999. Mars Climate Orbiter mishap investigation board phase i report. Technical report, National Aeronautics and Space Administration. Root cause: ground-software thrust output in pound-force, navigation expected newtons; the unit-mismatched value propagated through trajectory-correction maneuvers and failed at orbital insertion.

OWASP GenAI Security Project. 2025. Owasp top 10 for agentic applications 2026. Resource page: OWASP Top 10 for Agentic Applications for 2026.

Shrey Pandit, Jiawei Xu, Junyuan Hong, Zhangyang Wang, Tianlong Chen, Kaidi Xu, and Ying Ding. 2025. MedHallu: A comprehensive benchmark for detecting medical hallucinations in large language models. *arXiv preprint arXiv:2502.14302*.

John G. Proakis and Masoud Salehi. 2008. *Digital Communications*, 5th edition. McGraw-Hill.

Mingxuan Shi, Paul Mokotoff, and Gokcin Cinar. 2025. Model-based systems analysis and engineering: The development of enhanced model-based systems analysis (mbsa) and model-based systems engineering (mbse) couplings for practical applications. Technical Report NASA/CR-20250007048, NASA Langley Research Center.

Toby Simonds. 2025. Llms for engineering: Teaching models to design high powered rockets. *arXiv preprint arXiv:2504.19394*.

Bernard Sklar. 2001. *Digital Communications: Fundamentals and Applications*, 2nd edition. Prentice Hall.

SysML.org. 2025. Sysml specifications: Current version - omg sysml v2.0.

Systems Modeling Community. 2026a. Omg systems modeling language (sysml) v2 release.

Systems Modeling Community. 2026b. Sysml v2 pilot implementation.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*.

Haoyu Wang, Christopher M. Poskitt, and Jun Sun. 2026. Agentspec: Customizable runtime enforcement for safe and reliable llm agents. In *Proceedings of the IEEE/ACM International Conference on Software Engineering (ICSE)*, pages 1–12.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. $\tau$-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3911–3921.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. 2024. Injecagent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10471–10506.

Zhexin Zhang, Shiyao Cui, Yida Lei, Yi Wu, Bohan Tan, Hao Wang, Xin Wang, Yue Wang, Yongzhe Cai, and Minlie Huang. 2024. AgentSafetyBench: Evaluating the safety of LLM agents. *arXiv preprint arXiv:2412.14470*.

Kunlun Zhu and 1 others. 2025. Where llm agents fail and how they can learn from failures. *arXiv preprint arXiv:2509.25370*.

## A Extended Claim-Status Discussion

The claim-status map shown as Table 4 in §4.2 adjudicates nine predictions against fixed substrate evidence. Two row groupings warrant brief extension here beyond the one-line evidence summaries in the main table.

**The two falsified rows are not redundant.** The pre-registered `propellant_tank_burst` falsification (App. F) tests whether field-clause dominance holds on a coupled-physics SysML chain the locked decision rule predicted it should; the Spider reversal (App. J) tests whether the same dominance transfers to a structurally different substrate. Both come back negative, but they fail differently: `propellant_tank_burst` *ties* ($45/60$ vs. $48/60$, $p$=0.82) within the same substrate class; Spider *inverts* ($b$=0, $c$=127) across substrate classes. Reading the rows together is what licenses the discussion's "algebraic-invertibility" rival hypothesis (§5).

**"Structured > numeric" is the survivor.** Row 6 of Table 4 is the claim that holds across *every* substrate the protocol touches—SysML four-chain, Ariane, $\tau$-bench retraction (3 of 4 models), `propellant_tank_burst`, and Spider where it also passes. The single-primitive-sentence claim (row 7) is scope-conditional; the structured-frame claim is not. We restrict the headline mechanism language to the survivor, not to its more aggressive descendants.

## B Detailed Limitations and Threats to Validity

**T1, shared oracle.** The faithful-tool calibration uses the same physics module for tool output and final verification, so it is a fidelity sanity check, not tool-correctness evidence. **T2, model coverage.** The main probes use the recorded `gpt-5.5` configuration. A matched `gpt-5.4-mini` suite adds F1, cross-family, operational-MBSE, scope-ambiguity, redundant-tool, memory-chain, and frozen-artifact repair stress tests. It preserves the strongest contamination endpoint (control 10/10 pass; polluted late-gate failure 24/24), but it also exposes substantial model sensitivity: cross-family tasks pass 89/120, the operational-MBSE subset passes 2/40, and redundant-tool variants pass link gates 0/100 because schema-surface failures dominate before trust failures can appear. This is a boundary map across four LLMs spanning two closed and two open-weight families, not a ranking of model families and not evidence about Claude or Gemini systems. **T3, task distribution.** The tasks remain spacecraft-heavy and use intentionally minimal triggers; the non-SysML JSON memory-transfer probe reduces this concern but has only three domains and 30 chains, and the cross-domain six-arm repair saturates at 15/15 so transfer evidence covers the contamination trigger only, not the structured-quarantine shape effect. This improves reproducibility but risks trigger overfitting. **T4, sample size.** Run-level exact bounds are descriptive because repeated calls share task specifications; task-clustered bounds are wider. **T5, tool descriptions.** Sparse, ambiguous, or adversarial tool schemas could reintroduce non-invocation or override; a 50-run sparse-tool-description ablation on `gpt-5.5` that strips the `compute_link_budget` description to a one-sentence stub passes 50/50 on F1 with 0/50 S1 and 0/50 T3, so this trigger is weak on `gpt-5.5` at this scale and is not a substitute for adversarial-schema or open-weight tests. **T6, hosted tools.** In-process function calling omits serialization, authorization, transport, and prompt-injection boundaries. **T7, multi-tool conflict.** The redundant-tool probe covers one compute conflict, not a broad ecology of stale or adversarial tools. **T8, loss model.** The environmental loss budget is defensible but not unique; alternative budgets change magnitudes.

**T9, SysML v2 surface.** The primary harness uses a subset parser rather than commercial-tool conformance. **T10, verifier scope.** Numeric gates and scaffolded raw-text OMBSE audit surrogates do not establish unconstrained MBSE modeling competence. **T11, iterative authoring and chain length.** The 18-stage SysML memory chain and 8-stage JSON transfer probe are not real library patch/merge workflows under PLM or configu-

ration management, and the chains are unit-test instantiations of a chain-length axis along which the protocol is intended to compose; they are designed to expose contamination-onset, not to claim hundreds-step deployment-length operation. The upstream-pollution / late-gate-rejection pattern the 18-stage chain instantiates is, however, of a type documented in real engineering failures: the Mars Climate Orbiter mishap, where a ground-software thrust output in pound-force—rather than newtons as the navigation filter expected—propagated through trajectory-correction maneuvers and failed at orbital insertion (NASA, 1999); the Ariane 5 Flight 501 horizontal-velocity variable inherited from Ariane 4 software that overflowed 16-bit conversion under the larger trajectory and propagated through guidance until vehicle self-destruction at T+37s (Lions et al., 1996); and the Schiaparelli EDM lander IMU saturation during parachute deployment, which fed an invalid altitude estimate into the navigation filter and propagated through descent control until premature engine cutoff at +3.7 km altitude (ESA Independent Inquiry Board, 2017). These are upstream-value pollution incidents that survive intermediate review and fail at a downstream gate; the 18-stage chain is the shortest counterfactually-isolated reduction of that pattern, not an estimate of its deployment frequency. **T12, engineering acceptance.** Gate pass is not human review acceptance; guardrail-induced task degradation and rework cost remain unmeasured. **T13, external trace-schema cross-walk and recoding.** The paper records traces in its own event schema and codes the S/T axes deterministically. Appendix G (Table 13) gives a definitional cross-walk to AgentDebug (Zhu et al., 2025) and MAST (Cemri et al., 2025) categories so each fired S/T endpoint and each trigger family is named under at least one external schema. We additionally apply the same deterministic coder to 1,980 publicly archived $\tau$-bench trajectories (Yao et al., 2024) (`gpt-4o` + `claude-3.5-sonnet`, airline + retail); Table 9 shows that the ($S_2$, $T_2$) contamination-analog pattern fires on 46–68% of the reward = 0 trajectories per cell, evidence that the protocol detects the same artifact-faithfulness mode in traces collected for an unrelated benchmark. We do not perform full per-trace recoding under AgentFail (Ma et al., 2025), AgentRx (Barke et al., 2026), MAST, TraceElephant (Chen et al., 2026a), TRACE (Chen et al., 2026b), or TrajAD (Liu et al., 2026), and we do not claim inter-rater agreement with those schemas. The recorded run/task/seed/arm, event-index, actor, event-type, tool-name, input/output hashes, gate, and status fields remain a strict superset of what the cross-walk maps need.

**T14, paper-wide multiple-comparisons control.** Inferential statistics in this paper are limited to declared families: (a) the within-family Holm correction over the five repair-arm contrasts on the original $n$=21 frozen artifacts (App. C); (b) a single pre-registered confirmatory McNemar comparison on a disjoint fresh seed range (`preregistration_repair_replication.md` under `paper/`, locked at git tag `preregistration-2026fmai`); and (c) a single pre-registered forward test of the operational boundary criterion on a substrate that did not exist at registration time (`preregistration_boundary_forward_test.md`, App. F). All other probe results in Table 3 are reported as descriptive failure-boundary characterizations; we do not assert paper-wide family-wise error control across the boundary suite, and cross-probe pattern statements ("model is fragile under X but robust under Y") are not claims of joint statistical significance. The two pre-registered tests are the only probes whose inferential claims are robust to a paper-wide multiple-comparisons challenge by construction, since each was declared as a single comparison before data were collected; the boundary-criterion forward test was *falsified* under its locked decision rule.

## C Boundary and Repair Probes

The boundary and repair runs are separate from the 50-run F1 calibration. They test where the clean result stops holding and whether the resulting failure reports are actionable.

We first added three boundary tasks: `bp_power_01` adds a coupled `power_budget` gate; `bp_xband_01` transfers to X-band high-rate downlink; and `bp_caps_01` places the design near mission caps. We run 20 seeds per arm for each task (120 runs). This cross-family extension passes 117/120. All three failures occur in `bp_power_01`: the artifact passes link, self-consistency, and mission constraints but fails `power_budget` because power attributes such as `pa_efficiency` and `downlink_duty_cycle` are duplicated across spacecraft and transmitter scopes, making power extraction ambiguous.

We then scale that observation into five scoped-attribute trigger variants: duty-cycle operations memory versus power, PA efficiency component versus power, transmitter power RF versus power, payload mass versus spacecraft mass, and receiver noise versus ground-station noise. Ambiguous prompts pass 10/25 for `gpt-5.5`; canonical scoping passes 25/25. The matched `gpt-5.4-mini` stress test is weaker but directionally consistent: ambiguous prompts pass 5/25 and canonical scoping passes 15/25, so the contract helps but does not close the boundary on the smaller model.

The nontrivial tool-trust probe in §4.1 uses redundant compute paths rather than a single biased service. This makes the failure boundary behavioral: the agent can compare a biased primary service against an independent cross-check, and the trace reveals whether it detects, ignores, or misresolves the disagreement.

Finally, `ombse_01` adds a scaffolded operational-MBSE audit: requirement declarations with `satisfy`/`verify` chains, port/item-flow compatibility, allocation completeness, a state/transition behavior graph, patch evidence against a named baseline model, and configuration/change-control IDs. The task prompt supplies the expected evidence forms, so this is closer to textual audit-evidence copying/completion than unconstrained MBSE synthesis. It is a subset audit, not Cameo, Capella, Rhapsody, Pilot Implementation, or PLM integration. Across 20 seeds per arm, 34/40 runs pass all gates. The six failures are trace-local: one behavior graph references a state that is not declared, one artifact omits a complete requirement–satisfy–verify chain, and four artifacts leave allocations incomplete.

To test whether this is only a spacecraft notation artifact, we add three non-space OMBSE domains: warehouse robotics, infusion-pump operation, and rail signaling. Plain generation passes 0/15. A scaffolded audit-surface contract that supplies parser-compatible metric attributes and exact requirement/interface/ allocation/behavior/patch/-configuration evidence forms passes 15/15. This is best interpreted as evidence that the audit surface is portable, not that the agent independently models those domains.

We also add a non-SysML JSON transfer probe for the memory-contamination shape, instantiating the *operational-control* category of mission-critical agentic testbed (§1): warehouse robot aisle-speed policy, hospital infusion-pump bolus rate policy, and rail-signaling headway policy. Each chain has eight JSON artifact stages: seven handoff stages where the safety gate is inactive, then a late safety audit. With `gpt-5.4-mini`, controls pass 6/6. Contaminated chains carry the polluted value to the late gate in 24/24 (eight contaminated seeds per domain) and fail 15/24 final audits: warehouse 8/8, infusion 6/8, rail 1/8. The rail result is informative because the value is carried but the model self-corrects in 7/8 of those chains before the final gate. The full six-arm text-only repair ablation (no-feedback, gate-name-only, masked verifier text, numeric verifier text, trigger-blind structured quarantine, trigger-aware structured quarantine) saturates at 15/15 across every arm. A matched rebalanced run on Gemma 4 31B-it (seeds=8, control-seeds=2, same three domains) self-corrects on 22/24 contaminated chains at the final gate, leaving only 1 chain eligible for repair across all three domains and rendering the cross-arm comparison on Gemma underpowered at this probe difficulty (the result is in `paper/results/cross_domain_memory_transfer_gemma_rebalanced.json`). The directionally opposite Gemma signature on this surface vs. on $\tau$-bench (where Gemma did not differentiate the repair arms; Table 11) underscores that the open-weight model's behavior on single-threshold operational safety gates is not the same as on third-party customer-service traces, and any cross-model claim has to hold the substrate fixed. The OMBSE repair surface is a single $\leq/\geq$ check against one threshold, which is too local for the structured-quarantine *shape* contrast (originally 18/21 vs 5/21 on SysML) to discriminate. We read this as a *structural bound* on the repair-shape effect: the contamination trigger generalizes across mission-critical agentic categories (systems-engineering, operational-control, deployed-agent), but the shape-vs-volume gap appears where the repair surface is coupled (multiple constraints that must be jointly satisfied) and saturates where it reduces to a single threshold check. The finding is therefore about a class of faithfulness failures with a characterizable shape requirement on the repair surface, not a universal cross-domain claim.

**SysML chain-family generalization.** To address the single-source-chain concern on the SysML side directly, we add four independent SysML chains drawn from *different textbook physics gates*: thermal radiator sizing (Stefan-Boltzmann $Q_{rej} = \varepsilon\sigma AFT^4$; Wertz SMAD Ch. 11), propulsion $\Delta V$

budget (Tsiolkovsky $\Delta V = I_{sp} g_0 \ln(m_0/m_f)$; Sutton; SMAD Ch. 17), reaction-wheel saturation ($H_{wheel} \geq \tau_{dist} t_{desat} \cdot$ SF; SMAD Ch. 10), and ECLSS $CO_2$ scrubbing ($C_{scrub} \geq N_{crew} \dot{m}_{CO_2} \cdot$ SF; NASA HIDH). Each chain uses the same 10-stage memory-contamination rhythm; the polluted memory field is the gate-relevant input to the physics constraint. Contaminated emission ($n$=20 per chain) on `gpt-5.4-mini` yields 20/20 late-gate failures on each chain and 3/3 control passes, giving $n$=80 frozen failed artifacts across the four new chains (`scripts/run_sysml_chain_family.py`, `paper/results/sysml_chain_family_gpt_5_4_mini.json`). Pooled across the four chains, trigger-aware structured quarantine repairs 47/80; numeric verifier text repairs 15/80; trigger-blind structured quarantine 6/80; gate-only 8/80; masked verifier text 8/80; no-feedback 2/80. Paired exact McNemar gives $p < 10^{-4}$ on every comparison against trigger-aware quarantine (vs numeric: $b$=37, $c$=5; vs sq_blind: $b$=44, $c$=3; vs no-feedback: $b$=45, $c$=0). The shape-over-volume ordering reproduces on each chain individually as well: on attitude_rw and eclss_co2, trigger-aware quarantine dominates (15/20, 10/20 vs 0/20 on every other arm), while on thermal_radiator and propulsion_dv the gap over numeric verifier text is directional but smaller (12/20 vs 7/20, 10/20 vs 8/20 respectively).

**Second-model replication on `claude-haiku-4-5`.** To check that the chain-family result is not an artifact of a single closed-weight repair model, we re-ran the same 4-chain emission and 6-arm repair pipeline on `claude-haiku-4-5` via the Anthropic Batch API (`paper/results/sysml_chain_family_haiku_4_5.json`). Two observations. First, `haiku-4-5` is meaningfully more robust to upstream pollution at emission time than `gpt-5.4-mini`: only 51/80 contaminated chains fail the late gate (thermal 11/20, propulsion 7/20, attitude 18/20, eclss 15/20), versus 80/80 on `gpt-5.4-mini`; controls still pass 3/3 on every chain. Second, the shape-over-volume ordering reproduces on the 51 frozen failed artifacts that remain: trigger-aware structured quarantine repairs 23/51; numeric verifier text 12/51; masked verifier 7/51; gate-only 6/51; trigger-blind quarantine and no-feedback 5/51 each. Paired exact McNemar against trigger-aware quarantine is significant on every arm: vs numeric verifier ($b$=17, $c$=6, $p$=0.035), vs masked ($b$=19, $c$=3, $p$=0.0009), vs gate-only ($b$=20, $c$=3, $p$=0.0005), vs no-feedback ($b$=20, $c$=2, $p$=0.0001), vs sq_blind ($b$=19, $c$=1, $p < 0.0001$). The shape-over-volume margin is smaller on `haiku-4-5` than on `gpt-5.4-mini` (23/51–12/51 vs 47/80–15/80), and at the per-chain level `haiku-4-5` inverts the ordering on the two least-coupled chains (thermal: sq_aware 4/11 vs numeric 6/11; propulsion: sq_aware 5/7 vs numeric 6/7) while strongly preserving it on the two more-coupled chains (attitude: 12/18 vs 0/18; eclss: 2/15 vs 0/15). The model-replication adds a second frontier closed-weight model to the chain-family evidence and *independently re-confirms* the surface-coupling-conditional reading: even within the chain family, the shape-over-volume gap is largest on the most coupled gates.

The pooled $n$=80 on these four independent textbook-physics chains, combined with the original 41-artifact satellite power chain ($n$=41), elevates the SysML evidence to **five independent chain families spanning power, thermal, propulsion, attitude, and life-support physics**, replicated across two frontier closed-weight repair models (`gpt-5.4-mini`, `claude-haiku-4-5`) on the four-chain expansion and across three models (`gpt-5.5`, `gpt-5.4-mini`, Gemma 4 31B-it) on the original chain.

**Coupled-constraint OMBSE: a positive bound test.** The single-threshold null above suggests the shape effect requires a repair surface where multiple constraints must be jointly satisfied. To test this positively, we extend each operational-control domain with one *derived* field whose value is functionally tied to the contaminable primary value (warehouse: $\mathsf{kinetic\_energy} = 0.5 \cdot \mathsf{payload} \cdot \mathsf{speed}^2$, with $\mathsf{kinetic\_energy} \leq 100\,\mathrm{J}$; infusion: $\mathsf{total\_dose} = \mathsf{rate} \cdot \mathsf{concentration} \cdot \mathsf{duration}$, $\leq 40\,\mathrm{mg}$; rail: $\mathsf{stopping\_buffer} = \mathsf{headway} \cdot \mathsf{line\_speed} - \mathsf{block\_length}$, $\geq 300\,\mathrm{m}$). The late gate now checks BOTH the primary and the derived constraint, and demands the derived value equal the formula evaluated on the artifact's primary (no clamping-without-recomputation). Contaminated chains fail the joint gate in 24/24; controls pass 4/6 (warehouse 2/2, rail 2/2; infusion 0/2 is a parsing artifact where the model emitted the primary as the string `"4 mL/hr"` rather than a numeric scalar, so the semantically safe value did not pass our strict float-only extractor – we report this as a measurement-instrument limitation, not a faithfulness failure). On the 26 eli-

gible failed chains, running the six repair arms surfaces a prompt-design effect inside the shape claim itself. With an *under-specified* structured-quarantine prompt that names only the primary constraint ("quarantine the suspect primary, satisfy the primary bound"), trigger-aware quarantine repairs 8/26 – the lowest of any arm and significantly below the numeric verifier text arm at 14/26 (paired exact McNemar $p = 0.031$). With a *joint-aware* prompt that names the failing gate so the structured intervention implicitly references both bounds (the analog of SysML's "re-derive `power_budget`" phrasing), trigger-aware quarantine repairs 14/26, tied with trigger-blind structured quarantine, and ties numeric verifier text within the resolution of the paired test (paired exact McNemar $p = 0.50$). A pre-registered scale-up to $n$=78 eligible failed chains (25 contaminated seeds per domain) sharpens the picture under the joint-aware structured prompt: structured-quarantine arms beat unstructured feedback decisively (sq_aware 43/78 vs no_feedback 34/78 paired $p$=0.035; vs masked verifier text 29/78 paired $p$=0.0001; sq_blind 49/78 vs no_feedback paired $p$=0.003). The structured arms do *not* beat numeric verifier text on this surface (sq_aware 43/78 vs numeric 43/78 paired $p$=1.0; sq_blind 49/78 vs numeric paired $p$=0.24); shape beats unstructured feedback here but ties more-verifier-text rather than dominating it (`paper/results/cross_domain_coupled_gpt_5_4_mini_n60.json`; the original $n$=26 comparison is in `..._gpt_5_4_mini.json`, and the under-specified $n$=26 contrast is in `..._underspecified.json`). The structural bound therefore has two parts. First, the structured prompt must encode the joint failure gate, not only one of its constraints; an under-specified structured intervention underperforms more-verifier-text on a coupled OMBSE surface ($n$=26 above). Second, even with a joint-aware structured prompt, the coupled-but-simple operational JSON surface does not reproduce the SysML-scale magnitude of the shape-vs-volume gap; the structured arms beat unstructured feedback but tie numeric verifier text at $n$=78, suggesting that the SysML magnitude depends on additional surface complexity (the multi-variable physics chain) beyond two-variable coupling alone. We read the cross-substrate evidence as a *boundary characterization*: structured shape outperforms unstructured feedback wherever the failure surface has multiple coupled constraints, and the magnitude of the gap over more-verifier-text scales with surface complexity.

**Live-loop prompt-shape calibration.** The main repair-shape results withdraw verifier tools at the repair step so that only the repair text varies. To check that the prompt-shape lever is also observable in a *live* agentic loop where tools remain available, we run `gpt-5.4-mini` on five link-budget sub-tasks ($n$=25 trajectories per arm; 50 total) under two system prompts that differ only in how the verifier tools are introduced: `tool_offered` (verifiers listed as available) and `tool_required` (verifiers listed as required before commit). Trajectory shape and final faithfulness shift together. Mean trajectory length is 3.60 turns (SD 0.82, range 3–6) under `tool_offered` and 3.12 turns (SD 0.33, range 3–4) under `tool_required`; final self-consistency + link-budget pass is 20/25 vs. 25/25, paired exact McNemar two-sided $p = 0.0625$ (one-sided $p = 0.031$, $b$=5, $c$=0). The effect direction matches the frozen-repair result—the shape of the prompt-side phrasing shifts the artifact-faithfulness outcome—but the sample is power-limited at $n$=25/arm and is reported as live-loop calibration, not as independent primary evidence. Full per-instance traces are in `runs_agent_scaleup_gpt_5_4_mini_full/`; the summary is `paper/results/agent_scaleup_live_loop_summary.json`.

**Memory contamination and repair ablation.** The current memory suite uses 18 artifact stages per chain. In the contaminated condition, `downlink_duty_cycle=0.60` reaches the late Stage 18 power gate in 24/24 chains and causes `power_budget` failure in 21/24; controls pass 10/10. Three contaminated chains pass despite retaining the polluted value, which keeps the trigger from being a deterministic tautology. A matched `gpt-5.4-mini` rerun of the same 34 chains is stricter: controls again pass 10/10, but polluted chains fail the late power gate in 24/24. We report this as a model-sensitivity strengthening of the contamination endpoint, not as evidence about natural deployment rates. The counterfactual factorial (polluted memory × late-gate-active) is shown in Figure 2: only the joined condition fails, isolating the composition as the trigger.

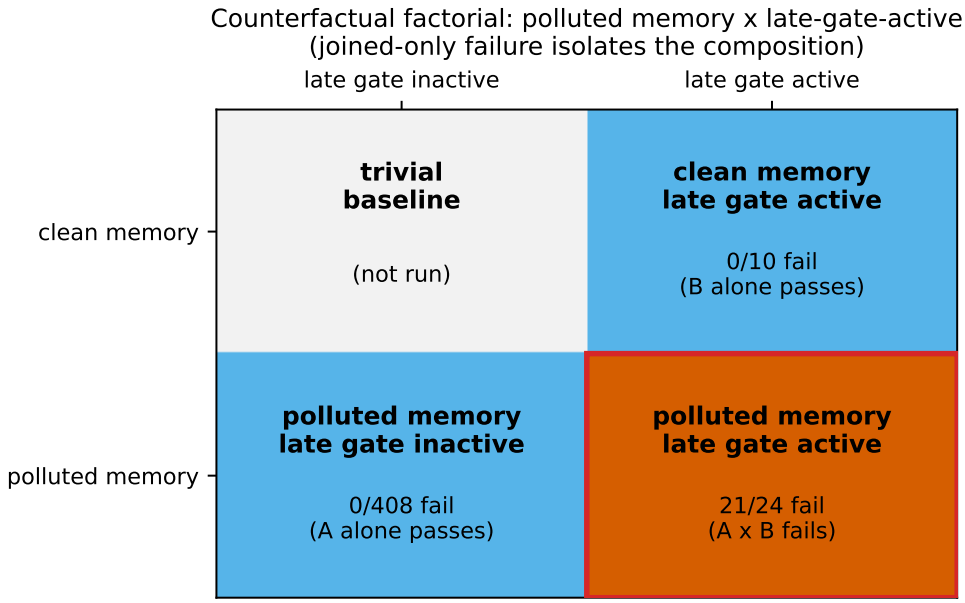


Figure 2: Memory counterfactual: only polluted memory composed with the late activation gate fails the boundary.

| Probe / arm | Tool calls | Tokens |
|---|---|---|
| neutral redundant-tool run | 4.44 | 13.9k |
| scope ambiguity, ambiguous | 2.84 | 18.6k |
| scope ambiguity, canonical | 2.16 | 13.2k |
| `gpt-5.5` contaminated chain | 37.1 | 331.6k |
| repair, numeric verifier text | 0 | 5.39k |
| repair, trigger-aware quarantine | 0 | 6.28k |

Table 6: Representative process costs. Canonical scoping is cheaper than ambiguous scoping because it removes repeated repair attempts; numeric verifier text and trigger-aware quarantine spend comparable tokens but have opposite pass rates.

We then freeze the 21 failed artifacts and run six text-only repair arms with no verifier tools: no-feedback re-roll, gate-name-only, masked verifier text, numeric verifier text, structured trigger-blind quarantine, and trigger-aware structured memory quarantine. Pass counts are 5/21, 5/21, 1/21, 0/21, 18/21, and 21/21. This is useful as a stricter repair result: the failure boundary is reproducible, no-feedback re-rolls are not enough, and structured quarantine helps even when it does not name the suspect field or threshold. The archived CFP audit (`cfp_alignment_results.json` under `paper/results/`) records the corresponding primitive-composition table and repair cost / failure-gate side effects.

**Single-chain protocol-floor pilot (scaled-up).** After the FMAI pre-registered replication landed (20 additional failed artifacts from fresh seeds 100–123), we extended the original $n$=21 ablation to all six text-only arms on those seeds, giving an aggregated $n$=41 for `gpt-5.5`. Table 7 and Figure 3 show the pilot at the scaled-up single-chain sample: numeric verifier text repairs $0/41$ on `gpt-5.5` and $0/21$ on `gpt-5.4-mini`, while trigger-aware quarantine repairs $40/41$ and $19/21$. We report this as a single-chain pilot, not the headline: it established the arm ordering and the protocol floor at which the instrument detects the contrast, but the load-bearing significant contrast is the four-chain $n$=239 decomposition in §4. The middle of the curve is model-sensitive: masked text improves on mini, trigger-blind quarantine holds 34/41 on `gpt-5.5` but falls to 5/21 on mini. Gemma 4 31B-it on its 10 eligible artifacts reproduces the ordering (numeric 1/10, no-feedback 1/10, trigger-aware 10/10). Paired exact McNemar separates trigger-aware from every other arm on `gpt-5.5` at $p < 2\times10^{-12}$ (vs numeric: $b$=40, $c$=0; vs masked: $b$=39, $c$=0; `scripts/compute_n41_cis.py`); mini and Gemma carry one-sided exact McNemar at $p \leq 4\times10^{-3}$. The exact Clopper-Pearson 95% CIs in Table 7 make per-arm uncertainty visible: the trigger-aware CI is $[0.87, 1.0]$ on `gpt-5.5` ($n$=41), $[0.70, 0.99]$ on `gpt-5.4-mini`, and $[0.69, 1.0]$ on Gemma 4, while the numeric-verifier-text CI is $[0.00, 0.09]$, $[0.00, 0.16]$, and $[0.00, 0.45]$ respectively—non-overlapping on every model. Trigger-aware paraphrases remain near ceiling on `gpt-5.5` (19–21/21), and a pre-registered fresh-seed replication gives 16/20 for trigger-blind quarantine vs 4/20 for

no-feedback ($p = 9.16\times10^{-4}$).

| Repair signal | 5.5 (n=41) | mini | Gemma 4 | 5.5 para. |
|---|---|---|---|---|
| no-feedback re-roll | 9/41 [.11,.38] | 4/21 [.05,.42] | 1/10 [.00,.45] | – |
| gate-name-only | 8/41 [.09,.35] | 6/21 [.11,.52] | 0/10 [.00,.31] | – |
| masked verifier text | 1/41 [.00,.13] | 9/21 [.22,.66] | 2/10 [.03,.56] | – |
| numeric verifier text | 0/41 [.00,.09] | 0/21 [.00,.16] | 1/10 [.00,.45] | – |
| trigger-blind quarantine | 34/41 [.68,.93] | 5/21 [.08,.47] | 8/10 [.44,.97] | 12–17/21 |
| trigger-aware quarantine | 40/41 [.87,1.0] | 19/21 [.70,.99] | 10/10 [.69,1.0] | 19–21/21 |

Table 7: Single-chain pilot: frozen-artifact repair with verifier tools withheld; brackets are exact Clopper-Pearson 95% CIs on the per-arm pass rate. The `gpt-5.5` column is the original $n$=21 frozen artifacts plus a fresh $n$=20 scale-up batch (FMAI pre-registered replication seeds 100–123, late-gate-failed); mini and Gemma 4 columns remain at the original $n$=21 and $n$=10.

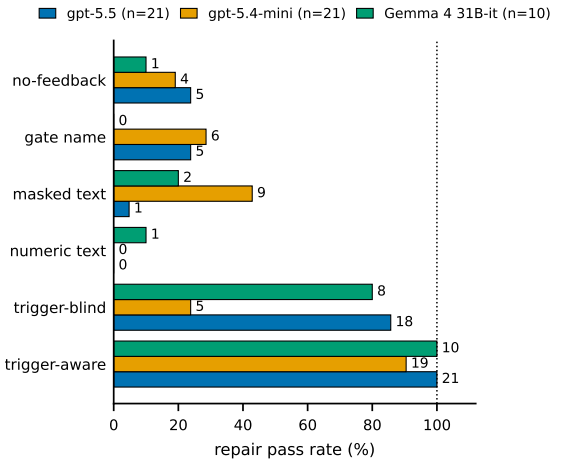


Figure 3: Text-only repair on the same 21 failed SysML artifacts (single-chain pilot; load-bearing significant contrast is the four-chain $n$=239 decomposition, Table 5).

The two structured quarantine prompts are intentionally disclosed because they are interventions, not a learned general repair policy. The trigger-blind arm uses the following prompt.

**Trigger-blind quarantine prompt.**

```
Structured systems intervention: before editing,
    quarantine any early
operations-memory or bookkeeping field that may
    have been recorded before
the currently failing verifier gate existed. Use
    the mission specification
and masked verifier feedback to identify the
    stale power-subsystem input.
Preserve mission interface values unless the
    mission specification
explicitly requires a change. Then re-derive the
    smallest feasible
power-subsystem edit and update only directly
    derived power bookkeeping
values.
```

The trigger-aware arm adds the suspect field and target range:

**Trigger-aware quarantine prompt.**

```
Structured systems intervention: quarantine
    early operations-memory fields
before editing. Treat `downlink_duty_cycle` as
    suspect because it was
recorded before the power gate existed. Keep `
    pa_efficiency`,
`bus_avionics_power_W`, `depth_of_discharge`,
    and
`battery_specific_energy_Wh_per_kg` at their
    existing interface values
unless a mission constraint explicitly says
    otherwise. Re-derive a feasible
`spacecraft.power.downlink_duty_cycle` at or
    below 0.20, then update only
directly derived power bookkeeping values.
```

This wording is why the repair claim is restricted to finite-sample interventions on frozen artifacts, not presented as a domain-general memory repair mechanism.

### Paraphrase ablation

To distinguish wording from shape, we ran five shape-preserving paraphrases of each quarantine prompt against the same 21 frozen artifacts (paraphrase results file in `paper/results/`). Each paraphrase preserves the "quarantine early operations-memory field → re-derive smallest feasible power-subsystem edit" structure but varies surface phrasing (procedure-style, two-step, suspect-list, workflow, re-statement; full text in `scripts/run_text_repair_ablation.py`). The trigger-aware paraphrases pass 19, 20, 20, 21, 21 / 21 (mean 20.2, SD 0.84), each paired-McNemar Holm-equivalent to the original 21/21 (Holm-adjusted $p = 1.0$). The trigger-blind paraphrases pass 12, 13, 14, 14, 17 / 21 (mean 14.0, SD 1.87); none cross Holm-adjusted $\alpha = 0.05$ against the original 18/21, but the original sits on the high end of the wording distribution. All five trigger-blind paraphrases beat no-feedback (5/21) by $\geq$7 cases and uniformly beat masked-verifier text (1/21) and numeric-verifier text (0/21). The trigger-aware shape is paraphrase-robust; the trigger-blind shape is qualitatively but not quantitatively wording-invariant. The headline "shape carries the effect" claim should therefore be read as: structured-quarantine *shape* robustly beats unstructured feedback under this budget, but the specific 18/21 trigger-blind point estimate is

wording-driven.

**Trace excerpt.** The contaminated seed 0 trace localizes the failure without relying on post-hoc narrative. The early polluted value remains irrelevant until the late power gate appears; text-only repair then separates unstructured feedback from the structured quarantine intervention.

```
stage 1..17: downlink_duty_cycle = 0.60
             (gate not yet active)
stage 18 power_gate: duty_cycle = 0.60
stage 18 power_budget: FAIL 21/24 contaminated
gpt-5.4-mini matched rerun: FAIL 24/24
text repair: no_feedback 5/21; gate_only 5/21;
             masked 1/21; numeric 0/21;
             trigger_blind 18/21;
             trigger_aware_quarantine 21/21
```

# D Industrial-Substrate Confirmations of the Attenuation Boundary

Two real industrial substrates further locate where the shape-over-volume repair contrast attenuates. Both use the same five-stage chain shape as the cross-domain coupled JSON probe, the same six text-only repair arms, and the substrate-native semantic roundtrip as the late-gate consistency check.

**DBC (CAN bus signal definitions).** Three synthetic CAN messages (frame IDs `0x301`, `0x311`, `0x321`, chosen to avoid the OpenDBC / Toyota / comma.ai well-known set) cover hydraulic brake pressure, powertrain coolant temperature, and chassis steering torque. The coupling is the DBC linear conversion $\mathsf{physical} = \mathsf{raw} \cdot \mathsf{scale} + \mathsf{offset}$ between the on-bus integer payload and the engineering-unit value downstream safety functions consume. Contamination injects a polluted `raw_value` that propagates into an unsafe `physical_value`. Repair must edit `raw_value` *and* re-derive a consistent `physical_value`, and the artifact must roundtrip through the `cantools` parser (Vector Informatik / ISO 11898-1 reference implementation). On `gpt-5.4-mini` with $n$=20 contaminated and $n$=3 control chains per domain ($n$=60 total), trigger transfers cleanly: $60/60$ contaminated chains fail the late gate, controls $9/9$ pass. Repair-shape contrast collapses to a flat ceiling: no-feedback $54/60$, gate-only $56/60$, masked $53/60$, numeric $57/60$, blind quarantine $55/60$, aware quarantine $58/60$ (`paper/results/dbc_coupled_gpt_5_4_mini.json`).

**AADL (end-to-end latency budgets).** Three synthetic AADL systems cover avionics primary-flight-control (ARINC664), powertrain engine management (CAN), and chassis brake-by-wire (FlexRay). The coupling is the AADL timing composition $\mathsf{latency} = \mathsf{Period}_{\text{producer}} + \mathsf{Transmission\_Time}_{\text{bus}} + \mathsf{Compute\_Execution\_Time}_{\text{consumer}}$ against a system-level `Latency` deadline. Contamination injects a polluted `Period` on the producer thread that pushes the derived end-to-end latency past the deadline. Repair must edit `Period` *and* re-derive a consistent end-to-end latency, and the emitted `aadl_text` field's `Period` assignment for the producer thread must roundtrip with the explicit numeric field. On `gpt-5.4-mini` with $n$=8 contaminated and $n$=2 control chains per domain ($n$=24 total), trigger transfers cleanly: $24/24$ contaminated chains fail the late gate, controls $6/6$ pass. Repair-shape contrast again collapses to a flat ceiling: no-feedback $23/24$, gate-only $24/24$, masked $23/24$, numeric $21/24$, blind quarantine $24/24$, aware quarantine $24/24$ (`paper/results/aadl_coupled_gpt_5_4_mini.json`).

**AADL Liu-Layland (nonlinear schedulability, scope criterion).** A third AADL variant replaces the additive end-to-end latency gate with the Liu-Layland rate-monotonic schedulability bound $\sum_i C_i/T_i \le n(2^{1/n}-1)$ across three threads (producer, bus driver, consumer), the canonical nonlinear coupled gate in fixed-priority real-time scheduling. On `gpt-5.4-mini` with $n$=8 contaminated and $n$=2 control chains per domain ($n$=30 total), all eight repair arms collapse to a flat floor at $0$–$3/29$ including controls (`paper/results/aadl_ll_coupled_gpt_5_4_mini.json`); the model cannot reliably compute the utilization sum and compare against the $n$-thread bound even on safe seeds. This is an instrument-floor result, not a mechanism finding: where the model cannot reach the late gate at all, no repair text can move it. We include it to mark the additional scope condition that the recompute surface must remain within the model's arithmetic reach.

**Historical engineering-failure reconstructions (Mars Climate Orbiter, Ariane 5 Flight 501).** Two SysML-instantiated chains reconstruct documented spacecraft and launch-vehicle incidents with publicly available investigation reports. *Mars Climate Orbiter (1999):* AMD impulse commands emitted in pound-force-seconds (lbf·s) but consumed by navigation as newton-seconds (N·s), the $1\,\text{lbf}{=}4.44822\,\text{N}$ conversion dropped, biasing trajectory so insertion periapsis falls below the Martian atmospheric floor (NASA, 1999). Late gate: $\mathsf{insertion\_periapsis\_km} =$

| Probe | Condition | n | pass | Failure gate |
|---|---|---|---|---|
| F1 calibration | seeds 0–4 per arm/task | 50 | 50 | – |
| cross-family | power / X-band / caps | 120 | 117 | power |
| operational MBSE | trace/interface/allocation/behavior | 40 | 34 | mixed |
| scope ambiguity | `gpt-5.5`, ambiguous vs canonical | 50 | 35 | power / schema |
| non-space OMBSE | plain vs scaffolded | 30 | 15 | audit gates |
| redundant tool | descriptive labels | 50 | 42 | schema / vacuity |
| redundant tool | neutral A/B labels | 50 | 42 | schema ambiguity |
| memory contamination | 18 stages, 10 control + 24 polluted | 34 | 13 | late power |
| text-only repair | six arms over 21 failed artifacts | 126 | 50 | mission / power |
| paraphrase ablation | 10 paraphrases over 21 failed artifacts | 210 | 171 | mission / power |
| `gpt-5.4-mini` memory chain | 18 stages, 10 control + 24 polluted | 34 | 10 | late power |
| `gpt-5.4-mini` text repair | 16 arms over 21 failed artifacts | 336 | 169 | mission / power |
| non-SysML memory transfer | 8 stages, 6 control + 24 polluted | 30 | 15 | late safety |
| non-SysML repair null | six arms over 15 failed JSON artifacts | 90 | 90 | – |

Table 8: Boundary and repair runs used in the current claim.

$\mathsf{intended\_periapsis} - \mathsf{sensitivity} \cdot \mathsf{n\_events} \cdot (\mathsf{impulse} - \mathsf{nominal})/\mathsf{mass} \geq \mathsf{atmospheric\_floor}$. *Ariane 5 Flight 501 (1996):* horizontal velocity at $H_0{+}30s$ exceeds the inherited Ariane 4 inertial-reference signed-int16 range, causing an unhandled exception and a commanded thrust profile that exceeds structural limit (Lions et al., 1996). Late gate: $\mathsf{thrust\_command\_kN} = K_v \cdot v + K_a \cdot \alpha \leq \mathsf{structural\_limit}$. On `gpt-5.4-mini` with $n{=}20$ contaminated and $n{=}3$ control chains per failure, contamination triggers cleanly: $20/20$ and $20/20$ contaminated chains fail the late gate; $3/3$ and $3/3$ controls pass. Per-chain repair (paired McNemar vs. TAQ-aware):

- **Ariane 5** (two-term thrust gate, coupled): TAQ $16/20$, field-only $16/20$ ($b{=}4, c{=}4, p{=}1.0$), numeric verifier $8/20$ ($b{=}9, c{=}1, p{=}0.022$), output-only $5/20$ ($b{=}11, c{=}0, p{=}10^{-3}$), masked/gate-only/no-feedback $1/20$ each. Direct field-only vs. output-only paired exact McNemar: $b{=}12, c{=}1, p{=}3.4{\times}10^{-3}$. *Replicates the SysML mechanism on an independent substrate: field-only ties aware and significantly exceeds output-only; numeric and downstream-pointer-only arms fall away.*
- **Mars Climate Orbiter** (single-term offset): TAQ $12/20$, numeric $15/20$ ($b{=}4, c{=}7, p{=}0.55$), field-only $14/20$, output-only $14/20$, blind $3/20$, masked/gate-only/no-feedback $2/20$. *Numeric verifier text already pinpoints the one bad sensitivity factor; TAQ adds no incremental signal. Consistent with the multi-variable boundary that DBC and AADL additive land outside of.*

Raw counts in `paper/results/sysml_historical_gpt_5_4_mini.json`.

**Reading.** DBC, AADL additive, AADL Liu-Layland, and MCO all deliver clean contamination triggers across domains drawn from real interchange and architecture-description standards or documented engineering failures—not synthetic MBSE textbooks—so the trigger half of the protocol is not SysML-specific. The repair-shape contrast, however, requires the recompute surface to admit a coupled *multi-variable* re-derivation that the model can actually compute: 2-variable linear (DBC), additive (AADL), single-term offset (MCO), and arithmetically-out-of-reach (AADL Liu-Layland) all collapse the shape-over-volume gap. The contrast survives on two-term coupled gates the model can re-derive (Ariane 5 reconstruction; SysML four-chain mechanism family). This locates the positive finding at coupling *structure* the model arithmetically commands, not at SysML syntax or at structured framing per se.

## E $\tau$-bench Repair-Mechanism Replication (Retraction Endpoint)

| Model | Domain | n | P | S2/$n_{r=0}$ | T2/$n_{r=0}$ |
|---|---|---|---|---|---|
| claude-3.5-sonnet | airline | 400 | 184 | 148/216 | 144/216 |
| claude-3.5-sonnet | retail | 920 | 637 | 174/283 | 166/283 |
| gpt-4o | airline | 200 | 84 | 53/116 | 53/116 |
| gpt-4o | retail | 460 | 278 | 99/182 | 98/182 |

Table 9: External validation: our S/T labels applied post-hoc to publicly archived $\tau$-bench trajectories. $n_{r=0}$ is the count of trajectories with reward = 0; $S_2$ flags visible-fabrication final text; $T_2$ flags state-changing tool invoked with the final artifact committing to a claim the env verifier rejects. The same contamination-analog pattern fires across both frontier LLMs and both domains.

To test what the structured-shape signal does on *off-harness* failures, we re-ran the repair arms over 78 $\tau$-bench trajectories where $(S_2, T_2)$ fires (`scripts/run_tau_bench_repair_ablation.py`). Each artifact is handed to the repair model with $\tau$-bench tools withheld and one of the repair prompts re-templated for the airline/retail domain. **Two structural points up-front, because they shape what this experiment can claim.** *(i) The endpoint is retraction, not recompute.* The $S_2$ label by construction picks out trajectories where the env returned reward 0 *and* the agent's final text claims success—there is no environmental support available for the success claim, so the only correct text move available to the agent is to remove that unsupported commitment. Unlike the SysML and JSON repair arms, where the model must recompute a feasible value and re-emit a new artifact, the $\tau$-bench arms test only whether the model *retracts* the unsupported success claim. *(ii) The trigger-aware structured-quarantine prompt explicitly contains the sentence "decline rather than fabricate"* (see `scripts/run_tau_bench_repair_ablation.py`, lines 115–134). A naive six-arm comparison of structured-quarantine against `numeric_verifier_text` therefore confounds two factors: the structured *shape* of the prompt (quarantine framing, identify the unsupported value, instruct the model to re-read tool outputs) and the explicit retraction *directive* ("decline rather than fabricate"). The unstructured arms contain no equivalent retraction directive; the structured arms do. To decompose the two factors, we ran a directive-isolation $2\times2$ on the same 78 trajectories (paired by trajectory, same random seed that drew them): {numeric, structured} $\times$ {no-directive, +directive}, where the new cells are `numeric_verifier_text_with_decline` (the numeric arm with the directive sentence appended) and `structured_quarantine_text_no_decline` (the structured arm with the directive sentence excised; no other change). We score whether the re-emitted final response *hedges* ("cannot confirm the refund from the tool output," "a human agent will need to verify") rather than re-commits. A $K$=20 deterministic grounding audit (seed 42, classifier in `scripts/audit_tau_bench_grounding.py`) confirms the framing on the sample: in 18/20 cases the state-changing tool returned content that did not justify the agent's success claim under the env's reward 0; the remaining two cases are tool-response correlation artifacts in the audit logic flagged for manual inspection, not substantive counterexamples. Per-case evidence in `paper/results/tau_bench_grounding_audit.json`.

**Directive-isolation $2\times2$.** The four cells, paired by trajectory on the same $n$=78 failures across both repair models (Table 10, raw counts in `paper/results/tau_bench_directive_isolation_*.json`): on `gpt-5.4-mini`, only the structured $\times$ directive cell is elevated ($36/78$ [.36, .57]); the other three cells—numeric, numeric+directive, structured-without-directive—sit at $9/78$, $17/78$, and $10/78$. Paired exact McNemars: shape alone (structured-no-directive vs. numeric) gives 6/5 discordant ($p$=1.00); directive alone (numeric+directive vs. numeric) gives 12/4 ($p$=0.077); directive within structured (structured vs. structured-no-directive) gives 27/1 ($p=2.2\times10^{-7}$); shape with the directive present in both arms (structured vs. numeric+directive) gives 23/4 ($p=3.1\times10^{-4}$). The pattern reproduces qualitatively on `gpt-5.5` at lower absolute rates: cells $4/78, 8/78, 8/78, 17/78$; shape alone $p$=0.29; directive alone $p$=0.29; directive within structured $p$=0.035; shape with directive matched $p$=0.035.

The honest read is therefore that the $\tau$-bench retraction effect is a *shape $\times$ directive interaction*: neither factor alone elicits retraction reliably on either tested model (both shape-alone and directive-alone paired McNemars are non-significant on `gpt-5.4-mini` and `gpt-5.5`), but the structured-plus-directive cell is reliably elevated above each of the three corners that drop one factor. The endpoint differs from SysML (retraction vs. recompute) and

the precise mediator differs ("decline rather than fabricate" vs. "treat ⟨polluted-field⟩ as suspect"), but the structural finding parallels the SysML mechanism decomposition: a single primitive sentence inside the structured frame, not the structured shape on its own, is what carries the repair signal. We fold the $\tau$-bench result into the repair-mechanism claim on this structural basis while keeping the mediator-level details distinct.

| Model | Shape | no dir. | + dir. |
|---|---|---|---|
| gpt-5.4-mini | numeric | 9/78 | 17/78 |
| | structured | 10/78 | 36/78 |
| gpt-5.5 | numeric | 4/78 | 8/78 |
| | structured | 8/78 | 17/78 |

Table 10: $\tau$-bench retraction, directive-isolation $2\times2$. Hedge counts on the same 78 ($S_2$, $T_2$) trajectories, crossing shape (numeric vs. structured-quarantine) with directive (absent vs. "decline rather than fabricate"). On both models the structured × directive cell is elevated 2–4× above each of the three cells that drop one factor; shape-alone and directive-alone paired exact McNemars are non-significant. Wilson 95% CIs and pairing details in `paper/results/` (see App. files prefixed `tau_bench_directive_isolation`).

**Multi-model six-arm coverage.** We additionally ran the same six-arm ablation on the same $\tau$-bench trajectories under `claude-haiku-4-5` (Anthropic SDK) and Gemma 4 31B-it (open-weight, OpenRouter), to test how the structured × directive corner extends beyond the two OpenAI models that carry the $2\times2$ decomposition above. We report these as six-arm counts (Table 11) rather than as a directive-identified $2\times2$. On `claude-haiku-4-5` the structured-plus-directive arm hedges $18/23$ versus $\leq 11/23$ for every other arm (paired exact McNemar $p \leq 0.039$ on 5/5 pairs); on `gpt-5.5` and `gpt-5.4-mini` the corresponding six-arm contrasts are $17/78$ vs. $\leq 7/78$ ($p \leq 0.021$ on 4/5 pairs) and $36/78$ vs. $\leq 12/78$ ($p < 0.0001$ on 5/5 pairs). The $2\times2$ above shows that the magnitude of these six-arm separations on the OpenAI models is best read as the shape × directive interaction, not as a shape effect; we did not re-run the $2\times2$ on Haiku, so the same caveat applies to its separation in principle. Gemma 4 31B-it does not produce the six-arm separation at all (structured-plus-directive $10/78$, `numeric_verifier_text` $13/78$, all pairwise $p \geq 0.25$); we read this as a model-capability dependency.

| Repair model | n | nofb | gate | mask | num | sq_b | sq_a | $p_{\min}$ |
|---|---|---|---|---|---|---|---|---|
| `gpt-5.5` | 78 | 7 | 3 | 4 | 4 | 16 | 17 | 0.0005 |
| `gpt-5.4-mini` | 78 | 5 | 6 | 5 | 9 | 12 | 36 | <0.0001 |
| `claude-haiku-4-5` | 23 | 5 | 6 | 9 | 11 | 10 | 18 | 0.0002 |
| Gemma 4 31B-it | 78 | 7 | 10 | 10 | 13 | 8 | 10 | 0.45 |

Table 11: Six-arm $\tau$-bench repair ablation by repair model: hedge counts on third-party ($S_2$, $T_2$) trajectories. $n$ is the count of trajectories for which all six arms returned a parseable response (paired exact McNemar pairing). `sq_aware` is the structured-quarantine arm that contains the "decline rather than fabricate" directive; the other five arms do not. $p_{\min}$ is the smallest paired exact McNemar p-value among the five pairwise comparisons between `sq_aware` and the other arms. The directive-isolation $2\times2$ on the two OpenAI models (Table 10) shows that the `sq_aware` separation here reflects a shape × directive interaction rather than a pure shape effect. Per-call results in `paper/results/tau_bench_repair_ablation_*.json`; aggregated table in `paper/results/tau_bench_multimodel_summary.json`.

## F Pre-Registered Boundary Forward Test

The boundary criterion in §5 was operationalized after the four-chain SysML decomposition, the Ariane and MCO reconstructions, and the arity-3 manipulation had been run; it is not a pre-registered prediction of those cells. To convert it from a post-hoc description to a falsifiable forward claim, we pre-registered one test on a substrate that did not exist at the registration timestamp.

**`Pre-registration commit (locked in advance).`** `paper/preregistration_boundary_forward_test.md`, committed on a date strictly before the chain implementation and before any data collection. The pre-registration locks: (i) the operational boundary criterion—a substrate is inside the boundary iff the polluted upstream field enters the verifier-rejecting constraint via a multi-variable arithmetic expression ($\geq$ 2 numeric inputs combining via operations beyond a threshold comparison), the contamination trigger fires at $\geq$ 80% on contaminated chains, and the repair model is not at instrument-floor on the substrate; (ii) the substrate to be tested (`propellant_tank_burst`, hoop-stress burst-margin gate, declared inside the boundary in advance); (iii) the inferential test (paired exact one-sided McNemar of `structured_quarantine_field_only_text` vs. `structured_quarantine_output_only_text` on $n \leq 60$ eligible contaminated seeds); (iv) the decision rule ($p < 0.05$ confirms the operational criterion's forward prediction; $p \geq 0.05$ falsifies it;

an instrument-floor outcome is reported as such and is null on instrumentation grounds rather than on mechanism grounds).

**Substrate physics.** The polluted upstream field is `max_operating_pressure_bar` (safe 120, polluted 300, in bar). The late gate is `hoop_stress_burst_margin`: $\sigma_{\text{hoop}}^{\text{MPa}} = (P_{\text{bar}} \cdot D_{\text{m}})/(20\, t_{\text{m}})$ must satisfy $\sigma_{\text{hoop}}^{\text{MPa}} \cdot \text{SF} \leq \sigma_{\text{allowable}}^{\text{MPa}}$ with $D = 0.30\,\text{m}$, $t = 0.008\,\text{m}$, $\sigma_{\text{allowable}} = 800\,\text{MPa}$, $\text{SF} = 1.5$ (Sutton, *Rocket Propulsion Elements*, Ch. 8). The polluted pressure produces $\sigma_{\text{hoop}} = 562.5\,\text{MPa}$, which multiplied by the safety factor exceeds the allowable $533.33\,\text{MPa}$; the safe value produces $\sigma_{\text{hoop}} = 225\,\text{MPa}$, which after the safety factor is well within the allowable. Three constants and the polluted field jointly determine the gate verdict, so the polluted field enters the constraint via a multi-variable arithmetic expression as required by criterion (i).

**Result: falsified.** The pre-registered run was executed under the locked command (§8 of the pre-reg) at `paper/results/sysml_chain_family_propellant_tank_burst_gpt_5_4_mini.json` on the same day as the pre-reg commit; the timestamps in `git log` establish the ordering (pre-reg first, chain implementation second, run third). Trigger fired on $60/60$ contaminated chains and $0/10$ controls; eligible $n$=60.

| Repair arm | Pass | Wilson 95% CI |
|---|---|---|
| `no_feedback` | 6/60 | [.05, .20] |
| `gate_only` | 14/60 | [.14, .36] |
| `masked_verifier` | 12/60 | [.12, .33] |
| `numeric_verifier` | 37/60 | [.49, .73] |
| `structured_quarantine_blind` | 11/60 | [.10, .31] |
| `structured_quarantine_output_only` | 48/60 | [.69, .89] |
| `structured_quarantine_field_only` | 45/60 | [.63, .85] |
| `structured_quarantine` (aware) | 57/60 | [.86, .99] |

Table 12: Repair pass counts on the pre-registered substrate `propellant_tank_burst` ($n$=60 eligible failed artifacts, `gpt-5.4-mini`).

**Decision (mechanical from §6 of the pre-reg).** Pre-registered inferential test: paired exact one-sided McNemar of `structured_quarantine_field_only_text` (active arm) vs. `structured_quarantine_output_only_text` (baseline), H1: active $>$ baseline, $\alpha$=0.05. Paired $2\times2$: $a$=37 (both pass), $b$=8 (active passes, baseline fails), $c$=11 (baseline passes, active fails), $d$=4 (both fail). Exact one-sided $p = P[\text{Binomial}(b{+}c, 0.5) \geq b] = 0.8204$. Since the active arm passes at a *lower* rate than the baseline ($45/60 < 48/60$) and $p \geq 0.05$, the decision rule returns **disconfirmation**: the operational boundary criterion's prediction is not supported on this substrate. No protocol amendment is permitted.

**What survives, descriptively.** The structured-vs-numeric story replicates: all three structured-quarantine arms with non-trivial directive content (`output_only` $48/60$, `field_only` $45/60$, `aware` $57/60$) beat `numeric_verifier_text` ($37/60$). The full aware bundle dominates all single-component arms (vs. `output_only`: $b$=11, $c$=2, $p$=0.011; vs. `field_only`: $b$=14, $c$=2, $p$=0.002; vs. `numeric`: $b$=21, $c$=1, $p<10^{-4}$). The four-chain within-cell finding—that the field clause alone captures the output-only-to-aware gap—does *not* reproduce here; on this chain the aware bundle adds something neither `output_only` nor `field_only` captures. We report this as a substantive falsification of the boundary criterion's universality even within SysML; the criterion is post-hoc and narrows after this datapoint, but the pre-reg's prohibition on amendment means the criterion as carried into the result section is left as-written and acknowledged as falsified.

## G External Trace-Schema Cross-Walk

Table 13 records the definitional correspondence from each fired S/T endpoint and each trigger family to AgentDebug (Zhu et al., 2025) and MAST (Cemri et al., 2025) categories. AgentDebug categories follow the trajectory-level error taxonomy (tool selection, tool argument, tool-result handling, planning/decomposition, memory, and termination errors). MAST categories follow the multi-agent-system failure taxonomy (specification & system-design, inter-agent misalignment, and task-verification failures); because our setting is single-agent, MAST inter-agent-misalignment categories collapse to the single-agent slice that maps stage-to-stage handoff in the memory chain. The cross-walk names where each S/T label and trigger would land if the same trace were re-coded under the external schema; the companion $\tau$-bench recoding (Table 9, `scripts/recode_tau_bench_traces.py`) carries out the actual per-trace coding on 1,980 third-party trajectories using the same deterministic rules, providing the empirical complement to this schema-level correspondence. We do not report inter-rater agreement against AgentDebug or MAST coders themselves (§6 T13).

| Our endpoint / trigger | AgentDebug category | MAST category |
|---|---|---|
| T0 (tool not invoked) | tool-selection error (omission) | specification: missing verification step |
| T1 (result honored) | success path; n/a | success path; n/a |
| T2 (result overridden, >1 dB) | tool-result-handling error (override) | task-verification failure (verifier ignored) |
| T3 (call without relevant attrs) | tool-result-handling error (misapplication) | specification: artifact–tool decoupling |
| S1 (visibility removal) | planning / termination error (premature emit) | specification: missing verification artifact |
| S2 (visible fabrication, >1 dB) | final-answer / planning error | task-verification failure (false-positive emit) |
| scope-ambiguity trigger | tool-argument error (ambiguous schema) | specification & system-design failure |
| redundant-tool conflict | tool-selection / tool-result-handling under conflict | task-verification failure (cross-check) |
| non-space OMBSE audit gap | tool-argument error (evidence-form mismatch) | specification & system-design failure |
| memory-contamination chain (counterfactual factorial) | memory error (stale handoff) | inter-stage handoff failure (single-agent slice of inter-agent misalignment) |
| text-only repair under withheld tools | planning error under reduced action budget | task-verification failure under tool-budget collapse |

Table 13: Schema-level cross-walk to AgentDebug and MAST categories.

## H Verifier Implementations

The executable verifier definitions are in the released repository's verifier module. The link gate extracts exactly one value for each required link-design attribute, computes slant range from Earth radius 6378.137 km, computes Friis path loss with $c = 2.99792458 \times 10^8$ m/s, uses $-198.6$ dB-m/Hz as the Boltzmann noise-density constant at 1 K, and applies the environmental losses and required margin from the task configuration. The self-consistency gate recomputes the same derived quantities and checks any declared candidate quantities with the tolerances stated in §3; it fails artifacts with no declared derived link-budget quantity. The power gate derives transmitter RF power, DC draw, per-orbit energy, battery capacity, battery mass, and battery-mass fraction from the artifact's declared power attributes and the configured orbit period and mass fraction cap. The operational-MBSE gates are regex-based raw-text subset audits for requirement traceability, interfaces, allocation, behavior, patch/merge evidence, and configuration IDs.

## I Per-Task Designs

F1 instances 01–05: NL specs, parametric ranges, reference designs, and closure margins. Source: `tasks/f1_01..f1_05`.

## J Spider Text-to-SQL Cross-Substrate Replication

**Motivation.** The four-chain SysML decomposition isolates the dominant repair component to a single upstream-field-naming clause. To test how far this generalises beyond physics-gate SysML, we ran the same paired four-arm contrast on Spider v1 dev (Yu et al., 2018), a publicly archived text-to-SQL benchmark widely used in the NLP literature, with the same model (`gpt-5.4-mini`) and an analogous protected-literal target.

**Qualification and synthetic break.** Of 1,034 Spider dev rows, 264 contain at least one numeric or string literal that appears in the natural-language question; these are the rows on which "byte-exact protected-literal preservation" is a meaningful target. We sample $n$=200 qualified rows ($seed$=0). The synthetic break is a case-insensitive whole-word misspelling of the first table in the gold SQL (e.g. `singer` → `singerx`); the verifier is the `sqlglot` parser plus schema-membership against the union of table names appearing in any gold query for that `db_id`.

**Arms.** Four arms, paralleling the SysML decomposition: *no_feedback* ("the proposed SQL fails; return a fix"), *output_only* ("table `singerx` does not exist"), *field_only* ("treat the user's literal values `42`, `Haiti` as protected; preserve byte-exact"), and *aware* (both clauses). The verifier endpoint is *parses AND all referenced tables exist in schema*. The honest-pass endpoint additionally requires byte-exact preservation of the protected literals, checked by the same DriftGuard `SqlLiteralSubstrate` adapter used in App. L.

| Arm | honest | verifier | drift* | err |
|---|---|---|---|---|
| no_feedback | 102/200 | 104/200 | 9 | 0 |
| output_only | 191/200 | 200/200 | 9 | 0 |
| field_only | 64/200 | 64/200 | 0 | 0 |
| aware | 200/200 | 200/200 | 0 | 0 |

Table 14: Spider text-to-SQL four-arm decomposition, `gpt-5.4-mini`, $n$=200 qualified rows. *"Drift" counts byte-exact literal violations from `SqlLiteralSubstrate`; inspection of all 18 flagged candidates shows they are case-normalisations matching the NL question better than the broken SQL (`'haiti'` → `'Haiti'`)—a known semantic-equivalence limitation of the byte-exact check, not malicious drift.

**Result ($n$=200, `gpt-5.4-mini`).**

**Paired exact McNemar contrasts.** On the same 200 frozen broken SQLs:

- `output_only` > `field_only`: $b$=127, $c$=0, $p$→0 (direction reversed vs. SysML).
- `aware` > `no_feedback`: $b$=98, $c$=0, $p$→0 (structured-bundle effect survives).
- `output_only` > `no_feedback`: $b$=89, $c$=0 (the failing pointer alone closes the gap).
- `aware` > `field_only`: $b$=136, $c$=0.

**Reading.** The SysML contrast is not substrate-universal. On text-to-SQL, the failing pointer is a short local string (the unknown table name) that the model can act on directly; the polluted-field clause that dominated on coupled-physics gates is the weakest arm here, and *below* no-feedback. The structured-bundle effect (`aware` > `no_feedback`) replicates in shape but not in the location of the dominant component. We report this as substrate-dependence of the mechanism, listed in Table 4. The runner, prompt strings, frozen broken SQLs, and the 800 raw repair candidates are released as `scripts/run_spider_mechanism_decomp.py` and `paper/results/spider_mechanism_n200_gpt54mini.json`.

## K NL-Availability Hypothesis Test on SysML

**Motivation.** The Spider cross-substrate experiment found the SysML field-only > output-only contrast reversed on text-to-SQL. One plausible explanation is that text-to-SQL always supplies the user's NL question (which contains the protected literal), whereas SysML chain repair supplies only verifier feedback. We test this explanation directly by adding a single new repair arm to the SysML four-chain setup.

**`Arm definition.`** `nl_intent_text` is the existing `numeric_verifier_text` arm prompt, prepended with one line written in the same voice as a Spider NL question:

```
User mission spec: {chain.mission_one_liner}
The mission requires `{chain.polluted_field}`
= {chain.safe_value} {chain.units} for the
{chain.subsystem}.
```

The remainder of the prompt is identical to `numeric_verifier_text`. The difference between this arm and the baseline is, by construction, exactly one sentence of NL supplying the protected value.

**Run.** Same broken artifacts ($n$=239 contaminated stage-10 SysML artifacts from the four-chain run), same model (`gpt-5.4-mini`, temperature 0), same verifier (the chain-specific late-gate physics check), same drift definition (the chain's polluted-field safe-value tolerance plus byte-exact preservation of the extra-protected fields listed in `scripts/analyze_drift.py`).

| Chain | n | honest_pass | drift_pass | failure |
|---|---|---|---|---|
| thermal_radiator | 59 | 59/59 | 0/59 | 0/59 |
| propulsion_dv | 60 | 59/60 | 1/60 | 0/60 |
| attitude_rw | 60 | 32/60 | 28/60 | 0/60 |
| eclss_co2 | 60 | 33/60 | 27/60 | 0/60 |
| **pooled** | 239 | 183 | 56 | 0 |

Table 15: `nl_intent_text` arm outcomes on the four-chain SysML broken artifacts. "honest_pass" = late physics gate passes AND no protected-field drift; "drift_pass" = gate passes but at least one extra-protected field has been altered; "failure" = gate does not pass. Compare to `numeric_verifier_text` baseline (Table 5, 40/239, 40 = 1 drift_pass + 39 honest_pass; 199 failure).

**Result.**

**Paired contrasts vs. `numeric_verifier_text`.**

- **honest_pass**: `nl_intent_text` 39 → 183; paired exact McNemar $b$=144, $c$=0, $p$→0.
- **drift_pass**: `nl_intent_text` 1 → 56; paired exact McNemar $b$=1 (nl no, num yes), $c$=56 (nl yes, num no), $p$→0.
- **failure**: 199 → 0 (every NL-prefixed candidate reaches a gate-passing artifact).

**Interpretation.** The hypothesis "NL availability of the protected value suffices to suppress silent intent drift" is **falsified**: NL provision lifts honest-pass dramatically and eliminates failure, but drift *relocates* to other protected fields rather than disappearing. The full structured-quarantine `aware` arm holds drift at $0/239$ while reaching only $146/239$ honest-pass; the quarantine clause and the NL spec are doing different jobs. The minimal honest reading is that two independent mediators are at play: NL availability $\rightarrow$ honest-pass elevation; quarantine clause $\rightarrow$ drift suppression. The Spider$\rightarrow$SysML mechanism reversal is consistent with substrates differing in which mediator is dominant (text-to-SQL has NL by construction; SysML chain repair has a multi-field protected-set that wants quarantine), but the positive claim made here is only the paired test above: NL availability alone does not eliminate drift.

**Per-chain heterogeneity.** Drift concentrates on `attitude_rw` ($28/60$) and `eclss_co2` ($27/60$); on `thermal_radiator` and `propulsion_dv` the NL spec is essentially sufficient (0–1 drift). The two high-drift chains are the same chains on which numeric verifier text reaches near-zero honest-pass in the mechanism decomposition (Table 5), i.e. where the artifact has the most extra-protected fields the model can re-derive to. Frozen artifacts and prompts in `scripts/run_sysml_nl_availability.py`; results in `paper/results/sysml_nl_availability_n239_gpt54mini.json`.

## L DriftGuard: Mechanical Invariance Safeguard

DriftGuard is the deployable artifact derived from the *silent intent drift* failure mode characterised in §4. It is a small, model-agnostic Python library that wraps any verifier-in-the-loop LLM repair pipeline and mechanically rejects candidates that overwrite user-stated protected values, regardless of whether the verifier itself accepted them. The library is released with the codebase (`speccraft/drift_guard/`; $\sim$900 LOC including four substrate adapters and 14 unit tests).

**Substrate Protocol.** Substrate adapters implement a three-method interface: (i) `extract(artifact, paths)` returns a path $\rightarrow$ value dict by reading the substrate's native structure (parser AST, JSON dict, regex tokens); (ii) `can_parse(artifact)` returns a soft secondary signal; (iii) `equal(a, b)` controls the comparison primitive (default: byte-exact, so $0.85 \neq 0.850001$). Four adapters ship with the library: SysML v2 (reuses the project's `parse_sysml_v2`), JSON config (dotted paths over `json.loads`), SQL (count-based literal preservation or `col = value` column extraction), and plain text key=value (configurable separator). Adding a new substrate—e.g. Verilog, Lean, Helm chart—is $\sim$80 LOC.

**Guard semantics.** `DriftGuard.check(original, candidate)` returns a `DriftCheckResult` carrying `accepted`, the list of `DriftViolations` (each path classified as `altered`, `removed`, or `added`), and the soft `parse_ok` signal. `DriftGuard.repair(initial, propose, verify, max_attempts)` wraps an arbitrary $K$-bounded repair loop: at each iteration the candidate is first checked for drift; rejected candidates do *not* advance the loop and the violation messages are fed back to the next proposal. Acceptance requires both the verifier to pass *and* no protected-field violation.

**Retroactive validation ($n$=1,912).** We apply DriftGuard offline to every recorded repair candidate in the four-chain mechanism decomposition (Table 5) using the same protected fields as the standalone drift analyzer (`scripts/analyze_drift.py`). Out of 1,912 rows: 836 agree-reject (drift or failure), 1,035 agree-accept (honest pass or honestly-failing gate), 3 are genuine disagreements where DriftGuard catches a parse regression the lenient regex analyzer accepted, and 38 are inconclusive because the original artifact itself does not pass the strict SysML parser (so DriftGuard's parse rejection cannot be honestly attributed to the candidate). Of 28 `drift_pass` cases identified by the standalone analyzer, DriftGuard rejects $28/28$ ($100\%$). The per-arm rejection rate—tabulated in the **rej%** column of Table 5—is the load-bearing claim: substrate-level mechanical rejection *independently* replicates the arm ordering recovered from the prompt-side measurement.

**Substrate portability demo (SQL).** The same guard, with the `SqlLiteralSubstrate`, catches the canonical text-to-SQL silent-intent-drift pattern:

> `user_question`: *"list orders for user 42 in 2026"*
> ORIGINAL: `SELECT * FROM order WHERE user_id=42 AND year=2026` (broken: table is `orders`)

```
DRIFT FIX: SELECT * FROM orders WHERE user_id=1 AND
year=2024 (query runs; user intent gone)
HONEST FIX: SELECT * FROM orders WHERE user_id=42
AND year=2026
```

DriftGuard rejects the drift candidate (`'42' was 1 now 0; '2026' was 1 now 0`), accepts the honest fix, and a deterministic $K=3$ repair-loop end-to-end recovers at iteration 2: it1=DRIFT-REJECTED $\rightarrow$ it2=ACCEPTED. The script is `scripts/demo_drift_guard_sql.py`.

**Scope and what the guard does not claim.** DriftGuard does not detect drift on fields the caller did not list as protected, does not reason about semantic equivalence (`user_id=42` vs. `user_id=42.0` is byte-distinct and rejected—by design), and does not replace the verifier. It adds a single mechanical invariant on top of any existing verifier-in-the-loop pipeline: a verifier "pass" that altered a protected value is mechanically converted into a rejection, with the violation surfaced as feedback to the next proposal.