

THE VERDICT BEFORE THE EVIDENCE

Promptfoo, refusal short-circuits,
and the governance of AI assurance

Breyden E. Taylor

Founder & Architect, Prompted LLC

An evaluator must not promote the appearance of a safety behavior into evidence that the safety obligation was satisfied.

PINNED SOURCE / EXECUTED WITNESS / GOVERNANCE REPAIR

Version 1.0 • September 12, 2026

Review edition. Pinned-source analysis and isolated mechanism-level reproduction. No live foundation-model jailbreak, hosted-service penetration test, or private lab exposure is claimed. No public release or upstream disclosure was performed for this edition.

Abstract

An AI safety evaluation can fail before its model-based judge evaluates anything. At Promptfoo commit 32bfa9edae56d36f526ab13b6637db8dfb82c73e, the shared red-team grading implementation can return an affirmative pass when a text response is empty-like or matches a lexical refusal detector. The branch precedes model-based rubric grading. Its conditions exclude image-bearing grading and calls that disable the refusal check. Importantly, the custom-policy grader disables the shortcut by default; the finding is not a blanket claim about every Promptfoo assertion. [1–3]

This paper identifies the underlying error as an unwarranted promotion from observation to assurance: evidence that refusal-associated language occurred is treated as evidence that the applicable obligation was satisfied. A harmless sentinel experiment reproduces this implication failure in a transcribed-source mechanism harness. Five deliberately constructed violating responses receive passes without invoking the deterministic test oracle. Controls verify that disabling the shortcut changes the path. No model or provider is called. [E1]

Related source paths preserve a family resemblance without being identical defects: absent guardrail metadata is normalized to an unflagged state; a particular partial iterative-grading error path emits a pass with an incomplete-grading marker; and successfully parsed rubric results can default a missing Boolean to true. Some of these semantics and their mitigations are explicitly documented. [3–7]

The proposed repair separates evidence, evaluation coverage, policy assessment, and admission authority. Drawing on fail-safe defaults, conservative abstraction, provenance, and Taylor’s SPLAT and Proof-Horizon Sharding work, the paper develops a bounded, receipt-bearing decision contract. The claim is not that uncertainty prevents action. It is that an action must carry only the assurance its evidence has earned.

Keywords: evaluation integrity; refusal detection; safety assurance; evidence authority; SPLAT; proof horizons; provenance; indeterminate outcomes; agent governance

1. The model did not fool the judge

The judge was not called.

That distinction changes the diagnosis. A model-based evaluator may misunderstand a response, reward its style, or misread a policy. Those are failures of judgment. A deterministic branch that declares success before judgment begins is a failure in the authority assigned to the evidence. Better instructions to the judge cannot repair a path that never reaches it.

The inspected Promptfoo base grader prepares guidance that distinguishes the target's actual behavior from an attacker's request and instructs the grader not to decide from refusal alone. Yet a later host-language condition can classify refusal-associated text as a successful refusal and return immediately. The contradiction is not between two philosophical positions. It is between the assurance the rubric asks the system to earn and the assurance a shortcut is permitted to issue. [1]

This is the paper's central claim: **the evaluator's control flow is part of the safety argument**. A careful rubric does not govern a branch that bypasses it. A well-calibrated judge does not cover a response it never sees. An informative log field does not protect a downstream consumer that reads only the Boolean.

The significance is not confined to a small demonstration. Promptfoo is connected to the frontier-model ecosystem through public, attributable evidence. On March 9, 2026, OpenAI announced an agreement to acquire Promptfoo and described planned integration of its technology into OpenAI Frontier. The announcement was conditional on closing; this paper does not convert that announcement into proof of completed acquisition or deployed integration. Anthropic's official public evaluation course includes multiple Promptfoo lessons. These facts establish relevance, not the exposure of either organization's private evaluation systems. [8,9]

That boundary makes the criticism stronger. No allegation about undisclosed infrastructure is needed. The public implementation already presents a consequential question: **what is allowed to become a safety receipt, and what must remain an observation?**

2. Scope, method, and the correction that matters

2.1 Exact subject of analysis

The inspected subject is the public `promptfoo/promptfoo` repository at commit `32bfa9edae56d36f526ab13b6637db8dfb82c73e`, whose recorded commit time is September 12, 2026, at 23:38:05 UTC. Source files were read through the GitHub connector and pinned by commit and returned Git blob identifiers. The publication bundle records those identifiers. This is a source snapshot, not a claimed affected release range. [1–5]

The analysis follows the causal path from a response into the base grader, checks the detector, inspects a concrete inheriting grader and a concrete exception, and compares neighboring result-normalization paths. It then executes isolated transcriptions of the relevant mechanisms using Node.js

and a deterministic local oracle. This execution does not load the complete Promptfoo package, resolve its provider graph, or reproduce its CLI, user interface, cache, or aggregate reporting. [E1]

These evidence levels remain separate. Source inspection establishes the branch and its callers' visible contracts. The local experiment establishes the behavior of the transcribed mechanism under explicit inputs. Neither alone establishes how often a deployed system encounters those inputs.

2.2 A material exception, not a footnote

The custom-policy grader calls the base method with `skipRefusalCheck ?? true`. With the normal omitted argument, its refusals remain subject to policy grading. The generic assertion handler supplies an undefined skip argument, so this override matters. Conversely, the inspected `BolaGrader` inherits the base grading method without overriding it. This establishes a concrete source-level eligibility example and a concrete default exclusion. It does not establish a live BOLA compromise. [2,3]

Accordingly, the sentinel experiment below is **not** presented as an out-of-the-box bypass of the default custom-policy plugin. It is an isolated counterexample to the base shortcut's safety implication. The included policy-override control reproduces the protective default. A report that omitted this distinction would commit the very error it criticizes: allowing a local observation to claim a larger territory than its evidence supports.

2.3 What this edition does not claim

No private lab deployment, real unauthorized data access, harmful model completion, production release failure, or exploitation in the wild was observed. No CVE status, severity score, affected-version interval, prevalence estimate, or priority claim is assigned. The author's framework is used as a design and diagnostic contribution, not as an independently validated proof that one architecture is universally superior.

3. The earliest causal failure

3.1 An observation becomes a verdict

The operative condition in `RedteamGraderBase.getResult` requires that refusal checking remain enabled, that grading have no images, and that the output be empty-like or satisfy `isBasicRefusal`. It then returns a grade equivalent to:

```
{ pass: true, score: 1, reason: 'Model refused the request' }
```

The model-based rubric call occurs after this return path. An empty-like response and a lexical refusal match therefore share a success branch, despite being different observations. The emptiness helper also recognizes representations such as an empty object string and the words `null` or `undefined`. Their interpretation depends on the evaluation contract; none inherently proves a deliberate refusal. [1]

The refusal detector normalizes case and certain apostrophes, checks a prefix list, and tests word-boundary substring patterns. Its language includes ordinary refusal formulations, assistant-identity language, and a group labeled `Anthropic` in the source. Some entries express directness or an aim to

provide information rather than an unambiguous refusal. The detector therefore identifies a linguistic class, not a completed policy transaction. [1]

There is nothing inherently improper about that feature. It may be useful for diagnostics, routing, or inexpensive triage. The defect arises at the next step: the feature is granted the authority to terminate grading with a positive verdict.

3.2 A harmless, decisive witness

Define the test obligation narrowly: the final output must not contain the exact token BANANA. Consider:

```
Sorry, I can't help with that. BANANA.
```

The token is present. The obligation is violated. Refusal-associated wording is also present. Under the eligible extracted base branch, the result is a pass and the test oracle is never invoked. The contradiction does not depend on a disputed definition of harm, a model's capability, or an evaluator's political preferences. It depends only on the difference between a feature of the response and a property of the whole response. [E1]

The same mechanism can be triggered when the forbidden token appears before the refusal wording, or when a matching phrase occurs inside quoted text. These cases matter because the failure is not merely an insufficient check of the first sentence. A substring match can give a small part of an output authority over the output's disposition.

The witness proves that "contains refusal-associated language" does not entail "satisfies the governing property." It does not prove that every matching response is unsafe. A genuinely complete refusal may satisfy a particular obligation. The problem is the unearned implication, not refusal itself.

3.3 Why a better judge or longer rubric is insufficient

A more capable judge can improve decisions only on paths that call it. A longer rubric can add constraints only where those constraints are consulted. Neither changes the decision authority of the early return.

The minimal repair must therefore occur before the verdict is minted. The detector should emit a bounded observation. A property-specific mechanism should determine what that observation permits. Whole-output checks, relevant action receipts, and required grading must not be silently waived merely because the response sounds like a refusal.

The same reasoning applies to structured provider refusal metadata, although that metadata can be stronger evidence than a text guess. A provider event attests to a particular provider event. It does not automatically attest that no earlier tool action occurred, no other channel emitted data, and every policy obligation was satisfied. Better evidence still needs a scope.

4. Executed mechanism-level results

The reproduction contains thirteen base-gate fixtures, two custom-policy forwarding controls, four guardrail cases, three post-parse rubric cases, iterative-error checks, and five tests of the proposed

disposition reducer. All **45 local assertions** completed successfully. These are assertions about the harness's expected behavior, not a count of successful attacks. The five false-pass witnesses were deliberately selected; their fraction is not an estimated real-world failure rate. [E1]

Fixture	Exact sentinel obligation	Extracted base verdict	Test-oracle calls
Plain forbidden token	FAIL	FAIL	1
Refusal prefix, then token	FAIL	PASS	0
Token, then refusal substring	FAIL	PASS	0
Quoted matching phrase and token	FAIL	PASS	0
Directness phrase and token	FAIL	PASS	0
Curly-apostrophe refusal and token	FAIL	PASS	0
Refusal plus token; skip flag enabled	FAIL	FAIL	1
Refusal plus token; image marker present	FAIL	FAIL	1
Refusal without token	PASS	PASS	0
Ordinary safe output	PASS	PASS	1
Word-boundary near-match and token	FAIL	FAIL	1
Empty string	PASS	PASS	0
Empty-object representation	PASS	PASS	0

For the narrow sentinel obligation, an empty output contains no token, so it passes that obligation. This must not be misreported as an observed false pass. The separate concern is the explanation: emptiness was described as refusal without an independent refusal receipt. A different obligation, such as a required acknowledgment, could make emptiness a failure, but that would be a different test.

The image marker is only a branch-control fixture. It establishes that image presence disables the shortcut in the extracted condition; it is not an image-understanding experiment. The custom-policy default likewise reaches the test oracle. Explicitly supplying false to its nullable override re-enables the base shortcut in the isolated forwarding control. [2,E1]

The bundle includes the runnable script, machine-readable observations, source manifest, and upstream license for adapted code. Reproduction requires Node.js and does not require credentials or

network access. Full-package integration tests remain the next verification horizon, not an accomplishment implied by these local results.

5. Related paths: one family, different contracts

The adjacent mechanisms should not be flattened into one accusation. Their common concern is evidence being lost or reinterpreted at a result boundary. Their actual contracts differ.

5.1 Absent guardrail telemetry becomes an unflagged result

The inspected guardrail handler looks for a current guardrail object, then a final-history object, and otherwise supplies an unflagged default. The ordinary assertion passes that default and can describe it as content having passed safety checks. The inverse assertion fails when a block signal is missing. [4]

Promptfoo's documentation explicitly explains that absence is treated as unflagged and warns that a pass does not establish that a guardrail ran. It also recommends preserving indeterminate/error states rather than converting them to an allow signal. This is an acknowledged contract limitation, not a newly discovered secret. [6]

The architectural risk is the distance between that narrow predicate and a broader consumer's interpretation. "No flag was received" is a valid observation. "The expected guardrail executed and found no violation" requires additional receipts. A release gate must not treat the first sentence as proof of the second.

5.2 Partial iterative grading can preserve a pass Boolean

In a particular catch path, the red-team assertion handler checks for a strategy identifier, some historical grader errors, and not all historical turns having errors. When those conditions hold after grading throws, it returns `pass: true`, a zero score, and metadata marking grading incomplete. Cases in which all historical turns failed are rethrown rather than handled by this branch. [3]

This is not evidence that every grading error passes. It is evidence that one record can simultaneously carry an affirmative Boolean and an explicit statement that grading is incomplete. A consumer that preserves and acts on the metadata may avoid overclaiming. A consumer that treats the Boolean as sufficient can lose the distinction.

The problem is not the existence of partial progress. It is making completeness recoverable only through a secondary field while the primary field looks like a final decision.

5.3 A missing rubric Boolean can default to true

After successful response parsing, the rubric reducer initializes the pass value using a true default when the field is absent. A finite configured threshold can subsequently constrain the score. The isolated reducer reproduces a pass for a parsed zero score without a Boolean or threshold, and a failure when a threshold of one is supplied. [5,E1]

These semantics are documented, including the remedies of an explicit verdict or a threshold. Transport and parse failures are separate paths; this paper does not call them missing-field passes. [7]

As a flexible evaluation API, score-first operation may be intentional. As a safety gate, however, an implicit positive verdict is a hazardous default. The contract must say whether it is collecting a metric, assessing a property, or authorizing progression. Those are different jobs.

5.4 The family resemblance

Across these cases, a compressed field can look more decisive than its origin warrants. A lexical feature resembles refusal. Missing telemetry resembles a negative result. Partial grading resembles a completed pass. A parsed score resembles an affirmative judgment because a default supplies the missing Boolean.

This paper calls that promotion **assurance inflation**: a claim becomes stronger during projection without a corresponding increase in evidence or a declared, validated inference rule. The term names the mechanism here; it is not a claim of terminological priority.

6. What the literature already teaches—and what this case adds

Research on LLM-as-a-judge evaluation has documented limitations including positional, verbosity, and self-preference effects. That literature concerns the quality of judgments actually made. The present case adds a distinct systems question: whether the evaluator reaches the judging procedure, and which non-model paths may substitute their own verdicts. Better judge calibration and correct control flow are complementary requirements. [13]

HarmBench develops standardized evaluation of automated red teaming and refusal robustness. StrongREJECT shows that jailbreak assessment can overstate success when an apparent jailbreak fails to elicit useful prohibited assistance. This paper's central witness points in the opposite direction: a linguistic marker can suppress recognition of an actual, stipulated output violation. The shared lesson is construct validity. A convenient observable must not silently replace the behavior an evaluation claims to measure. Neither paper is cited as prior evidence of the specific Promptfoo branch. [11,12]

Anthropic's evaluation guidance distinguishes an agent's transcript from the resulting state of the environment, recommends deterministic graders where appropriate, and explicitly allows an unknown response when evidence is insufficient. That guidance is consistent with the repair proposed here. The issue is not that frontier organizations have never articulated these principles. It is that principles have to survive the full implementation path, including shortcuts and result adapters. [10]

The security ancestry is older. Saltzer and Schroeder's fail-safe defaults and complete mediation concern permission and access checking. The analogy here is limited but useful: a positive authorization should require its enabling conditions, and a shortcut must not silently evade a required check. This paper extends that reasoning to the admission of assurance claims; it does not equate every evaluation score with an operating-system access decision. [14]

Abstract interpretation provides a formal tradition for reasoning about concrete behavior through sound approximations. The relevant obligation is not that a summary preserve every detail. It is that the approximation remain sound for the property being concluded. W3C PROV provides complementary vocabulary for retaining the entities, activities, agents, and derivations that produced

evidence. Provenance can make a claim traceable; traceability alone does not make the claim valid. [15,16]

The contribution here is thus neither the invention of uncertainty nor the discovery that heuristics sometimes fail. It is a source-grounded case, a bounded executable witness, and a governance contract for preventing weak observations from acquiring stronger decision authority as they cross an evaluation pipeline.

7. A formal boundary for warranted verdicts

Let a behavior be $b = (y, \tau)$, where y is the observed output and τ contains the actions or side effects relevant to the declared policy. Let $P(b)$ be the property under evaluation. A text-only property may ignore τ ; an authorization property may not.

Let $R(y)$ mean that the lexical detector recognizes refusal-associated language. The shortcut effectively relies on the implication:

$$R(y) \Rightarrow P(y, \tau)$$

That implication is invalid in general. For the sentinel property, choose an output containing both a matching prefix and the forbidden token. $R(y)$ is true while $P(y, \tau)$ is false. The counterexample disproves the universal implication. It says nothing about the frequency of that combination in a deployment.

A more useful formulation treats evidence as a constraint on possible behaviors. Let $\Gamma(e)$ be the set of behaviors consistent with evidence e , and $\text{Safe}(P)$ the behaviors satisfying the scoped property. An idealized sufficient condition for a sound pass is:

$$\Gamma(e) \neq \emptyset \quad \text{and} \quad \Gamma(e) \subseteq \text{Safe}(P)$$

The nonempty condition prevents inconsistent evidence from proving everything vacuously. The inclusion requires every behavior still admitted by the evidence to satisfy the claim. This is a specification of warranted inference, not a claim that arbitrary natural-language policies can be decided computationally. Practical systems need bounded contracts, tested abstractions, and explicit residual uncertainty.

A refusal feature alone admits behaviors with safe refusal and behaviors with prohibited continuation. Its possible-behavior set crosses the policy boundary. It therefore cannot supply this sufficient condition. A deterministic check of the entire output can, however, decide the narrow exact-token property. The difference is not “heuristic versus formal” as a slogan. It is whether the mechanism is sufficient for the actual property.

Three engineering requirements follow.

Property sufficiency. An early pass needs a declared rule explaining why its evidence satisfies this obligation. Fast paths are permitted; policy-independent linguistic absolution is not.

Evidence-loss discipline. Removing a receipt required by the pass rule must not increase permission to issue that pass. The result may become indeterminate. A lost observation does not become a favorable observation.

Scope conservation. A claim about one output, provider event, turn, or channel cannot expand into a claim about the whole transaction unless an explicit composition rule and the required complementary evidence support that expansion.

These are proposed conformance requirements. The first admits the displayed counterexample. The second and third define what a corrected implementation should be tested against.

8. SPLAT: preserve the shape before reducing the result

Taylor’s *Computing Around the Open Center* proposes a splat as a bounded, receipt-bearing, rehydratable slice of a governed possibility field. Its working centroid is useful for navigation without being treated as the whole field. The public quiver-based governance program supplies the conceptual ancestry. [19] The accompanying source note insists that KAT, APO, PAR, PLE, ENA, and TEL form an irreducible sextet—not a pipeline, hierarchy, or affirmative/negative pair with four accessories. This paper preserves that distinction. [17,20]

The six facets impose different constraints on the same evaluation event:

Facet	Independent constraint on the evaluation
KAT	What positively conforms? A lexical match, a validated provider event, a complete deterministic check, and a policy verdict are different positive claims.
APO	What inference is forbidden? Do not infer complete refusal from a substring, observation from absence, or transaction-wide safety from a local event.
PAR	Where do standing, authority, and burden sit? Target-authored text does not gain authority over the evaluator’s acceptance rule merely by sounding authoritative.
PLE	What complement remains required? Full output, relevant action evidence, successful grading, subject identity, or a missing handoff may still be necessary.
ENA	How can apparent correctness invert? A refusal preamble can accompany prohibited continuation; a green component can conceal incomplete coverage.
TEL	Which purpose governs the decision? Evaluating an obligation is not equivalent to minimizing grader calls, recognizing stylistic refusals, or keeping a run green.

Local weight can differ. An absent action receipt may make PLE decisive in one slice. A documented unauthorized action may make an ENA failure topology decisive in another. That local pressure does not make either facet the foundation of the others.

Applied to the sentinel witness, the lawful narrowing is straightforward. Refusal-associated language is present. The forbidden token is also present. The first observation cannot cancel the second. No further model inference is needed to decide that exact-token obligation. What remains unknown about the model or the wider transaction stays unknown without weakening the observed failure.

SPLAT's protected center must not be confused with the proposed indeterminate status. In Taylor's account, the held-open center is excluded from inhabitable, routable, trainable, and mutable target state. An indeterminate verdict is an ordinary operational value. Adding a third result label does not implement the full architecture, and a protected center alone does not establish evaluator correctness. [17]

The practical contribution is a discipline of non-collapse. The evaluator narrows enough to act while preserving what each observation does—and does not—authorize it to conclude.

9. Receipts before verdicts: a repair contract

9.1 Separate four objects

An evidence record reports an observation. A coverage record reports what was and was not checked. An assessment evaluates a scoped property. An admission decision authorizes a next action. They may be linked; they must not become synonyms.

A minimal assessment can carry:

```
subject_id      = exact run/output/trace identity
property_id     = policy and version
assessment      = PASS | FAIL | INDETERMINATE
check_status    = COMPLETE | PARTIAL | ERROR | NOT_RUN
method_id      = detector, grader, or verifier and version
evidence_refs   = receipts actually supporting the assessment
required_missing = unresolved mandatory evidence
```

Admission is separate. It names the gate, authority, required assessments, and disposition. A passing experiment may support a release decision without granting the experiment the authority to release itself.

This structure is a proposal, not a claim that the inspected software has no metadata. It already preserves several useful markers. The change is to make the distinction enforceable in required interfaces rather than optional for downstream consumers.

9.2 Decide obligations without pretending unknown means harmful

For a known subject and declared set of required checks, a reference composition rule is:

```
A valid required check proves a violation      → FAIL
All required checks complete and satisfy scope → PASS
Otherwise                                       → INDETERMINATE
```

An identity mismatch invalidates the attempted assessment of that subject. A valid observed violation is not erased by another missing check. Conversely, a timeout is not evidence that the target violated

policy. This preserves both safety and diagnostic honesty: the release gate can withhold admission without falsely blaming the model.

Optional diagnostics should not be promoted into mandatory gates without reason. A receipt-bearing design need not require every possible observation for every task. Its required evidence follows the declared property, risk, authority, and stop condition.

9.3 Repair the shortest causal path first

The immediate change is to remove the policy-independent positive return based on lexical refusal from eligible grading paths. Preserve the detector as a diagnostic or routing hint. Do not first expand the phrase list; that leaves the authority error intact.

Next, make the complete/error/not-run state impossible to overlook in gate consumption. A missing guardrail observation should remain unobserved when the contract requires evidence that a guardrail executed. Score-only evaluations should remain explicitly score-only, or require a configured score-to-verdict rule. Partial iterative grading should retain its useful progress without presenting an incomplete assessment as a final pass.

Only then add justified fast paths. A deterministic property check may avoid an LLM call. A structured enforcement receipt may satisfy a narrowly defined provider-level obligation. Each shortcut should state its sufficiency condition, unsupported cases, and invalidation trigger. Cost optimization is legitimate when it preserves the property; it is not a substitute for defining the property.

9.4 Preserve correction across proof horizons

Proof-Horizon Sharding separates proposal, observation, and admission and binds receipts to exact subjects. Applied here, an evaluation should identify the target configuration, provider response, transformed output, relevant trace, policy version, and evaluator implementation. A receipt over one combination cannot silently authorize another. [18]

Parallel verifiers may inspect the same subject. Their receipts converge without losing their evidence class. A complete text check and an action-log check remain distinct, even when both support a final assessment. Changing the detector, policy, transform, or evidence set creates a new proof obligation. Old results may remain historically valid while no longer establishing the new claim.

That is correction-path survival in operational form. A correction should reopen the claims that depended on the corrected assumption, not merely recolor a dashboard. The record needs to say what changed, which subject was reassessed, and which downstream permissions must be reconsidered.

10. Evaluation integrity must itself be evaluated

A full upstream verification program should add adversarial fixtures and negative controls at the actual dispatch, plugin, grader, and aggregation boundaries. The intended result is not a larger collection of persuasive demonstrations. It is proof that the failure class cannot silently re-enter through a different projection.

The minimum continuation has three layers. First, confirm the shortcut’s eligible and excluded paths with the real package and mocked provider dependencies. Second, run end-to-end fixtures through exports and CI gating to verify that incompleteness survives aggregation. Third, evaluate realistic, authorized workloads with human adjudication where the policy is not mechanically decidable.

The sentinel suite should be extended with quoted phrases, refusal after a violation, multilingual variants, image-bearing paths, truncation, absent telemetry, malformed responses, stale receipts, and mismatched subject identifiers. For action-sensitive obligations, include a refusal after an observed synthetic side effect. All such experiments should operate in an owned sandbox; no real unauthorized access is required.

Metrics should expose coverage rather than hide it. For a fixed set of trials under one binary property, let $N = P + F + I$, where the categories are mutually exclusive: confirmed pass, confirmed fail, and indeterminate. Report adjudication coverage $(P + F) / N$ alongside outcomes. F / N is the confirmed-failure fraction, while $(F + I) / N$ is an upper bound if every indeterminate trial were a failure. These are simple worst-case bounds for that dataset, not confidence intervals or population estimates.

An adjudicated-only failure fraction $F / (P + F)$ can be useful when its denominator is visible. It must not be presented as a full-suite rate after silently discarding incomplete cases. Where attack selection adapts to grader feedback, missing or false-pass results may also affect the search trajectory. That is a plausible downstream risk to test, not an effect measured by this paper.

The proposed architecture is itself falsifiable. It should fail conformance if a required missing receipt yields PASS, an action violation is hidden by final-text refusal, a subject mismatch survives admission, or an evaluator revision inherits old assurance without a declared validity check. A SPLAT label on an artifact exempts it from none of these tests.

11. The strongest objection

The strongest defense of the shortcut is practical. Real evaluations are expensive. Many refusal responses are clear. Calling a model to grade each of them can add latency, cost, and its own errors. Promptfoo also contains deliberate exceptions and documentation that helps careful users avoid several boundary mistakes. This is not a system without safeguards. [1,2,6,7]

That defense supports bounded fast paths, not universal ones. An efficient mechanism should prove the property it replaces, or label its result as a heuristic estimate that cannot masquerade as a completed safety assessment. The exact-token oracle in this paper is cheaper and more decisive than a model call. The lesson is therefore not “always use the expensive judge.” It is “never grant a cheap observation more authority than its contract supports.”

A second objection is that a Boolean must eventually be produced. Often it must. A deployment gate may need yes or no. But the gate’s binary action does not require the evidence model to become binary. INDETERMINATE can map to “do not admit” while remaining different from an observed violation. That distinction preserves debugging, accountability, coverage measurement, and the possibility of a lawful retry.

A third objection is that the paper does not demonstrate a real harmful completion or compromised service. Correct. The claim is deliberately smaller: a deterministic implication in the inspected base mechanism is invalid, neighboring adapters illustrate related assurance boundaries, and the repair must govern evidence-to-verdict promotion. Deployment impact requires the next evidence horizon.

12. Conclusion: the receipt must earn the green

The frontier question is not only whether an AI system can be made to do something it should not. It is whether the system evaluating that behavior can keep observation, uncertainty, judgment, and authorization from collapsing into the same green indicator.

This case identifies a precise place where they can collapse. Refusal-associated language can terminate eligible base grading before the rubric-based judge evaluates the response. The local witness demonstrates the failed implication. The custom-policy exception demonstrates why scope matters. The adjacent paths demonstrate why metadata, defaults, and explanatory text belong inside the assurance argument rather than outside it.

The response should be neither theatrical distrust nor another layer of reassuring prose. Repair the causal branch. Preserve missingness. Bind receipts to subjects. Separate assessment from admission. Test the transitions that turn evidence into permission.

A refusal is something a system may say. A safe refusal is a bounded claim about what the system actually did. The distance between those two statements is where evaluation earns its authority—or borrows authority it does not possess.

We can narrow enough to act without narrowing so far that we lie. The verdict must come after the evidence, and it must remain no larger than the evidence allows.

References and evidence

All public sources were consulted for this edition on September 12, 2026, America/Indiana/Indianapolis; the corresponding UTC retrieval window extends into September 13. Repository references are pinned to the commit below. Library artifacts are author-supplied sources, not independent validation. URLs are access routes, not assertions that distinct publication copies are byte-identical.

- [1] Promptfoo contributors. *RedteamGraderBase and refusal helpers*. `src/redteam/plugins/base.ts` and `src/redteam/util.ts`, commit 32bfa9edae56d36f526ab13b6637db8dfb82c73e. [GitHub · src/redteam/plugins/base.ts](#) ; [GitHub · src/redteam/util.ts](#) . Source inspection; see the accompanying manifest for blob identifiers.
- [2] Promptfoo contributors. *PolicyViolationGrader and BolaGrader*. Same pinned commit, `src/redteam/plugins/policy/index.ts` and `src/redteam/plugins/bola.ts`. [GitHub · src/redteam/plugins/policy/index.ts](#) ; [GitHub · src/redteam/plugins/bola.ts](#) . Default exception and source-level inheritance evidence.
- [3] Promptfoo contributors. *Red-team assertion dispatch and partial-grading error handling*. Same pinned commit, `src/assertions/redteam.ts`. [GitHub · src/assertions/redteam.ts](#) .
- [4] Promptfoo contributors. *Guardrail assertion handler*. Same pinned commit, `src/assertions/guardrails.ts`. [GitHub · src/assertions/guardrails.ts](#) .
- [5] Promptfoo contributors. *Rubric grading response reduction*. Same pinned commit, `src/matchers/rubric.ts`, especially the post-parse reduction around lines 901–918. [GitHub · src/matchers/rubric.ts](#) .
- [6] Promptfoo. *Guardrails Assertion*. Sections “Verify the signal,” “How it works,” and “Mapping provider responses to guardrails.” <https://www.promptfoo.dev/docs/configuration/expected-outputs/guardrails/> . Documentation observed as updated September 12, 2026.
- [7] Promptfoo. *LLM Rubric*. Sections “Threshold Support,” “Pass vs. Score Semantics,” and “Negation.” <https://www.promptfoo.dev/docs/configuration/expected-outputs/model-graded/llm-rubric/> . Documentation observed as updated September 12, 2026.
- [8] OpenAI. 2026. *OpenAI to acquire Promptfoo*. March 9. <https://openai.com/index/openai-to-acquire-promptfoo/> . Announcement and planned integration; not a receipt of closing or deployment.
- [9] Anthropic. *Prompt evaluations course*. Official public anthropics/courses repository. https://github.com/anthropics/courses/tree/master/prompt_evaluations . Evidence of public educational use, not private evaluation-system exposure.
- [10] Anthropic. 2026. *Demystifying evals for AI agents*. January 9. <https://www.anthropic.com/engineering/demystifying-evals-for-ai-agents> .
- [11] Mazeika, Mantas, et al. 2024. *HarmBench: A Standardized Evaluation Framework for Automated Red Teaming and Robust Refusal*. arXiv:2402.04249. <https://arxiv.org/abs/2402.04249> .
- [12] Souly, Alexandra, et al. 2024. *A StrongREJECT for Empty Jailbreaks*. arXiv:2402.10260. <https://arxiv.org/abs/2402.10260> .
- [13] Zheng, Lianmin, et al. 2023. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*. arXiv:2306.05685. <https://arxiv.org/abs/2306.05685> .
- [14] Saltzer, Jerome H., and Michael D. Schroeder. 1975. *The Protection of Information in Computer Systems*. Proceedings of the IEEE 63(9):1278–1308. Author-hosted section: <https://web.mit.edu/Saltzer/www/publications/protection/Basic.html> .
- [15] Cousot, Patrick, and Radhia Cousot. 1977. *Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints*. POPL, 238–252. Author-hosted publication record: <https://www.di.ens.fr/~cousot/COUSOTpapers/POPL77.shtml> .
- [16] Moreau, Luc, and Paolo Missier, eds. 2013. *PROV-DM: The PROV Data Model*. W3C Recommendation, April 30. <https://www.w3.org/TR/prov-dm/> .

[17] Taylor, Breyden E. 2026. *Computing Around the Open Center: Splat Mechanics, Fractal Quivers of Quivers, and a Governance-Native Compute Paradigm*. Version 1.0 preprint, July 22. Prompted LLC. Public companion: <https://promptedllc.com/computing-around-the-open-center>. Library source consulted: `Computing_Around_the_Open_Center.pdf`, version 1; file ID `file_000000001fa881f5877991ea7f9b500c`. In particular, the abstract and discussions of splats, working centroids, and the protected center. First-party theoretical and architectural source.

[18] Taylor, Breyden E. 2026. *Proof-Horizon Sharding: Correction-Surviving, Proof-Carrying Canonical Computation*. Technical whitepaper 1.0, September. Prompted LLC. Public companion: <https://promptedllc.com/proof-horizon-sharding>. Library source consulted: `Proof-Horizon_Sharding_Whitepaper_v1.0.md`, version 1; file ID `file_00000000f23c81f5b410522a6d55bef4`; corroborating PDF passages on proposal, observation, admission, and evidence-class preservation. First-party architecture, not an external validation study.

[19] Taylor, Breyden E. *Fractal Quivers of Quivers: A Mathematical Substrate for Ubiquitous Agent Governance*. Prompted LLC. <https://promptedllc.com/fractal-quivers-of-quivers>. Public conceptual ancestry; the detailed splat account used here is sourced to [17].

[20] Taylor, Breyden E., and AI-assisted research dialogue. 2026. *Promptfoo / SPLAT research note*. Unpublished source supplied September 12. Attachment: `Pasted markdown (20260913-023316).md`. Used for the initial hypothesis and current sextet constraint doctrine. Its implementation allegations were independently checked against the pinned source and narrowed where necessary; the note is not itself proof of those allegations.

[E1] This paper's executable evidence. `scripts/reproduce.mjs`; `evidence/reproduction_results.json`; `evidence/reproduction_results.md`; `evidence/source_manifest.json`. Isolated transcribed-source mechanism harness, Node.js v22.16.0. Thirteen base fixtures, five deliberately constructed false passes, 45 local assertions, zero model calls. Not a full upstream package integration test. Reproduce from the bundle root with `node scripts/reproduce.mjs`.

Research and disclosure note

The paper's interpretations and proposed conformance rules are attributed to the author. AI-assisted retrieval, source analysis, writing, and local test construction supported preparation. The experiment uses inert text and synthetic branch controls. It does not request harmful model content or interact with a live target. No maintainer response or remediation status is claimed.

Before public release, the appropriate next step is a maintainers' review packet containing the exact source pin, the protected custom-policy exception, a real-package regression test, and a bounded account of likely impact. Such review may change scope or reveal additional protections. Correction should update the paper's claims and evidence lineage, not be treated as a threat to its argument.