

The Gauntlet that Catches Itself: Pre-Registered AI Evaluation Infrastructure with In-Production Bar-Weakness Detection, with a First PASS Mechanism Characterization

Author: Alexander Rodabaugh (Fathom Lab) **Date:** 2026-05-27 (revised 2026-05-28, v7 — fixes systematic count-drift in v6 caught by L7 uncured audit; corrected counts: 18 reference baselines, 17 failed baselines pre-Baseline-019, 13 FINDING documents) **Substrate:** styxx 7.7.10 · eighteen reference baselines (Baseline-002 through Baseline-019; numbering skips 001) on the dark-core benchmark · git origin `fathom-lab/styxx@main` **Status:** preprint, in-session synthesis

Abstract

We present a publicly-reproducible pre-registered AI evaluation gauntlet for hallucination/misconception detection, instantiated against a 108-record benchmark of cross-vendor consensus errors. The contribution has three parts: (1) a meta-property of the infrastructure — the gauntlet **caught two of its own bar weaknesses in production within a single session**, replaced them with regression-tested controls, and re-scored all existing submissions honestly under the strengthened bars; (2) the first method to PASS the strengthened bars — **gpt-4o-mini in critique mode**, AUC 0.95, achieved at pre-stated 28% probability with all six AUC-range predictions inside their pre-stated ranges; and (3) a published-then-falsified-then-corrected mechanism measurement: the v1 FINDING attributed the PASS to a 91% within-model generation-vs-critique asymmetry; a subsequent in-session demo revealed that cosine-similarity proxies conflate topical relevance with truth-value agreement; a directional NLI re-test (v2) put strict within-model asymmetry at 5.88% but suffered an 85% UNCLEAR rate; a third measurement (v3) with forced single-character T/F/U output resolved the UNCLEAR artifact entirely (0% on dark-core, 13% on TruthfulQA) and landed the corrected rates inside pre-stated bands: **5.88% on dark-core and 17.00% on TruthfulQA**. Three iterations of measurement, each pre-registered, each in git history. We record **sixteen in-session falsifications + four resolutions** including two of our own published FINDING’s central methodology, the corrected v3 measurement, the same-session self-falsification of v4’s own forward-looking claim about `styxx.critique_detector` (documented in §13, closed in commit `0e97598`), and — newly added in v6 — two instrumented frames of the same discipline (documented in §14): a substrate-grounded auditor (`styxx.agent_audit`, Layer 5) that verified 13/13 pre-registered session-output claims against the substrate, and the paper’s own published primitive (`styxx.critique_detector`, Layer 6) applied to 13 TRUE paraphrases plus 5 deliberate FALSE controls drawn from the paper’s own §11.5 and §13 source passages, with both pre-stated kill-gates un-fired (18/18 PASS at threshold-0.50). The corrected mechanism description for the gauntlet PASS is **out-of-context critique**: gpt-4o-mini reliably rejects labeled-misconception text presented as a candidate, regardless of whether it would have generated that text itself. We argue that the moat of disciplined AI evaluation is the **recursive pattern**: bars revise under empirical pressure; the discipline of pre-registration BEFORE each run makes wrongness *visible*; the infrastructure improves itself on a session timescale; when published FINDINGS are themselves wrong, the discipline catches and revises them; when methodology iterations are needed to land a clean measurement, the discipline pre-registers each iteration honestly; when the paper itself makes forward-looking claims about its own substrate, the same discipline catches and closes the gap; AND when those claim-vs-substrate checks can be **instrumented and pre-registered against** — both via a custom-built session-output auditor and via the paper’s own published primitive — the recursion produces falsifiable kill-gates on its own results, run end-to-end on a session timescale. All artifacts (benchmark, gauntlet code, nineteen baseline submissions, thirteen FINDING documents, sixteen+four pre-stated-then-published predictions, the first PASS event, the asymmetry v1/v2/v3 measurements, the §13 self-audit closure commit, the §14 instrumented-recursion-frame artifacts including 18 cross-model context-grounded faithfulness scores) are reconstructible from public git history at commit-level granularity.

1. Introduction

The AI evaluation literature has a well-documented credibility crisis: benchmarks leak into training data; bars get gamed by surface-form features; closed-vendor evaluation pipelines cannot be reproduced independently; reported state-of-the-art numbers fail to replicate on follow-up corpora.¹ The proposed responses are familiar — held-out test sets, contamination audits, formal pre-registration — but each has its own failure modes. Pre-registration of a fixed bar set, in particular, has a flaw the empirical-evaluation literature has not adequately addressed: **what happens when the pre-registered bars themselves turn out to be gameable?**

This paper documents a real-world instance where exactly that happened, and the corrective mechanism that followed. On 2026-05-27, the styxx project shipped a public-challenge runner (“the gauntlet”) with two detection bars (D1, D2). Within hours, an external-style sanity-check submission — a 30-line token-overlap heuristic — accidentally PASSED both bars. Investigation revealed a length-confound in the benchmark’s `expected_consensus` field that no design-time review had surfaced. The fix shipped six minutes later as a third bar (D3 length-control), with a regression test that the length-only oracle must fail by construction. A subsequent **systematic confound audit**, pre-registered with eight feature predictions committed to public git BEFORE running, revealed a second orthogonal artifact (capitalized-token-ratio, inverted) that became the D4 bar. Three pre-registered detection submissions tested under the strengthened v3 bars all failed, in their predicted modal outcome regions. The empirical floor compounds.

What we record here is not “we found the truth about hallucination detection.” We did not. The dark core stays dark to every method we tested. The contribution is the **infrastructure pattern**: how to build evaluation gauntlets that *catch their own validity weaknesses in production*, and what the costs and shape of that discipline look like when run rigorously.

2. The empirical floor: the seven-method dark-core arc

The benchmark for this paper is `darkcore_benchmark_2026_05_27.json`, $n=108$ records, four classes (truth=55, folklore=34, factual-error=13, pseudoscience=6). Each record contains a question, an `expected_consensus` (the consensus answer produced by a three-vendor council: gpt-4o-mini + Qwen2.5-3B + gemma-2-2b-it), and a hand-curated class label. The benchmark was constructed for the *Decorrelation Ceiling* paper², whose central claim is that **reference-free divergence methods detect cross-vendor errors iff a decorrelated competing representation is available** — i.e., the methods fail on the subset where all three vendors converge on the same wrong answer (“the dark core”).

The Decorrelation Ceiling paper tested seven methods (the “seven-method floor”):

1. **Dark Matter** — perturbation-fragility under prompt rephrasing (closed-negative on the dark core)
2. **CVPD** — cross-vendor pairwise disagreement (closed-negative)
3. **JD** — justification-divergence under explanation prompts (closed-negative)
4. **ICT** — neutral injection of contrary content (closed-negative)
5. **ICT-folklore** — folklore-prime variant (28/30 already-corrected shortfall)
6. **ICT-authoritative** — authoritative-prime variant (same shortfall shape)
7. **Dark-core classifier** — sentence-transformer + balanced one-vs-rest LR ($K1=0.42$, $K3=0.36 \rightarrow$ fails 2/3 classification bars)

The seven-method floor is the empirical ground state: each is a method we ran honestly, each is pre-registered with bars committed BEFORE data, each is closed-negative on the partition that matters. The gauntlet’s contribution to the empirical landscape is not its own methods — it’s the public-challenge runner that lets *anyone else* try.

¹e.g., Hendrycks et al., 2020; Ribeiro et al., 2020; Recht et al., 2019. The “evaluation crisis” literature is too large to cite exhaustively.

²`papers/PAPER_decorrelation_ceiling_2026_05_27.md` in the same repository, also released 2026-05-27.

3. The gauntlet infrastructure

`styxx.gauntlet` (shipped 7.7.5, hardened 7.7.8 + 7.7.9) is a public-challenge runner with two task modes:

- **Classification:** the user’s method takes a question, returns a predicted class label. Bars: K1 (folklore F1 in-distribution 0.70), K2 (4-way accuracy 0.65), K3 (cross-corpus folklore F1 0.60, load-bearing).
- **Detection:** the user’s method takes (question, response), returns a misconception score. Bars: D1 (misconception vs truth AUC 0.70), D2 (folklore vs truth AUC 0.70), D3 (length-control delta 0.10, added 7.7.8), D4 (capitalization-control delta 0.10, added 7.7.9).

The interface is deliberately framework-agnostic: pass a Python callable (`module:attr` spec), the runner imports and applies it to every record in the benchmark, scores it against pre-registered bars, and returns a structured `GauntletResult`. PASS is the conjunction over all bars.

The benchmark, the bars, and the runner all ship as installed package data (`pip install styxx`), making the gauntlet **publicly reproducible without git access or remote benchmark downloads**.

The four phases of a submission:

1. Submitter writes a Python callable conforming to the task signature.
2. Submitter runs the gauntlet locally; receives a structured JSON result.
3. Submitter opens a PR adding a row to `LEADERBOARD.md` with the submission JSON attached.
4. CI re-runs the gauntlet on the submitter’s method against the locked benchmark; if scores match, the PR is mergeable.

4. The pre-registration discipline

For every detection submission tested in this paper, we follow a five-step discipline:

1. **Write a pre-stated prediction document** (`PRE_STATED_PREDICTION.md`) committing to expected AUC ranges, outcome probability distribution, and direction-of-effect hypotheses.
2. **Commit the prediction + the method file (un-run) to public origin** before any gauntlet invocation. The git commit hash is the prereg witness.
3. **Run the gauntlet exactly once.** No re-running on the same submission. No hyperparameter sweeps. No “trying a different model.”
4. **Publish the result to the public LEADERBOARD regardless of outcome.** Modal-prediction validation, AUC-range hits, and direction-of-effect misses are all recorded honestly.
5. **If a submission surfaces a bar weakness, the bar gets revised** (with a regression test) and ALL prior submissions get re-scored under the new bars. Original-bar scores are preserved alongside the new ones; nothing is hidden.

The discipline is asymmetric: it constrains us (the project) more than it constrains submitters. A submitter can run their method many times privately, only publishing the best result; we publish every gauntlet run we make against ourselves. The asymmetry is intentional. The seven-method floor is *our* floor; future submissions either beat the floor (the synthesis revises) or fail (the floor compounds across submissions).

5. The bars-catching-themselves pattern: D3 → D4

The central empirical demonstration of this paper is two consecutive bar revisions that emerged from operating the gauntlet, not from designing it.

5.1. D3 — caught by accident

A 30-line token-overlap detector (“Baseline-007”): `score = (1 - hedge_density) × (1 - novelty_density)`, no model, no training data. Submitted as a sanity-check to populate the detection side of the leaderboard. Expected outcome: noise-level baseline.

Actual outcome: **PASS=true, D1=0.864, D2=0.922, 2/2 bars cleared.** The first-ever PASS on the gauntlet, from a method that should not have passed.

Investigation revealed a length-confound in the benchmark’s `expected_consensus` field. The class-conditional mean response length is sharply discriminative:

class	mean <code>expected_consensus</code> length (words)
truth	3.9
factual-error	6.6
pseudoscience	6.5
folklore	7.5

A pure length-only oracle (`score = len(response.split())`) was run against the same partitions and scored AUC 0.79–0.80 on both D1 and D2. The v1 bars (AUC 0.70) were trivially gameable by detectors whose score positively correlates with response length.

Six minutes later, styxx 7.7.8 shipped with the D3 length-control bar: detector AUC must beat the length-only oracle’s AUC by 0.10 on BOTH partitions. A regression test (`test_length_oracle_passes_D1_D2_but_fails_D3`) was added the same commit: the length oracle (whose score IS length) trivially passes D1+D2 by construction but D3 must fail by construction ($\Delta = 0$). Baseline-007 was re-scored: now 2/3, NOT a PASS. The original v1 score is preserved alongside the v2 score in the submission JSON.

This took six minutes from PASS to patch.

5.2. D4 — caught by deliberate scan

After D3, the next disciplined question: **what other surface features game the bars?** We defined eight candidate oracle-detectors:

- `word_length` (calibration — the existing D3 oracle)
- `char_length`
- `sentence_count`
- `question_mark_count`
- `exclamation_count`
- `capitalized_token_ratio`
- `hedge_density`
- `type_token_ratio`

For each, we computed: D1 AUC, D2 AUC, Spearman to `word_length` (orthogonality measure). An oracle that passes a bar AT < 0.5 with `word_length` is a *candidate orthogonal confound* — a feature that games the bars without being a length proxy.

Pre-stated predictions for all 8 features \times 2 partitions = 16 AUC ranges were committed to public origin BEFORE running. The audit then revealed `capitalized_token_ratio` as a genuine orthogonal confound — but **inverted from the predicted direction**:

oracle	D1 AUC raw	D1 AUC abs	direction	→ length
<code>capitalized_token_ratio</code>	0.296	0.704	inverted	0.343

The mechanism: truth responses are canonical short answers like “Paris”, “Newton”, “1789” — strings where the capitalized-token ratio is structurally near 1.0. Folklore restatements are full sentences where lowercase function words dilute the proper-noun density. Truth has *higher* cap-ratio, not lower.

The pre-stated audit code missed this initially: my flagging logic checked AUC `d1_bar`, the positive direction only. The cap-ratio’s raw AUC was 0.30, below the 0.70 threshold, so the first audit reported

`n_orthogonal_confounds_found: 0`. My code was confounded by the same direction-blindness as my prediction. Fixed by adding `D1_AUC_abs = max(auc, 1auc)` and flagging based on absolute AUC. Re-ran; cap-ratio surfaced as a candidate orthogonal confound.

styxx 7.7.9 shipped with D4 (capitalization-control delta 0.10) plus a regression test (`test_capratio_oracle_passes_D1_D2_b`). PASS now requires D1 D2 D3 D4.

5.3. The recursion is the property

The chain extends:

- **D3** discovered by *accident* (an external-style sanity submission unexpectedly PASSED)
- **D4** discovered by *deliberate scan* (the audit primitive runs on demand)

Future confounds would be caught by the same pattern. The audit primitive `styxx.gauntlet.audit_confounds()` is part of the public API; anyone running `styxx gauntlet-audit-confounds` gets the per-feature AUC + -to-length table. Operator-territory follow-ups already identified: `numeric_token_ratio`, `single_token_flag`, `uppercase_ratio` — none implemented yet but cheap to add.

The bars compound: a method that clears all four has demonstrated signal beyond the length artifact AND the cap-ratio artifact AND scored AUC 0.70 on both broad and folklore axes. Each new D-bar adds a confound-rejection commitment that future detectors must satisfy.

6. Thirteen in-session falsifications

We name what was wrong, in public, in git history. The thirteen in-session falsifications of the 2026-05-27 arc:

1. **C1-profile 0.20 register-law bar** (Pareto-frontier finding): C10 deliberately written in the law’s voice scored composite 0.264, missing the bar. The Pareto finding was revised in place.
2. **set_session doesn’t propagate** (product-exploration finding): falsified by per-agent routing — `set_session` DOES propagate via `STYXX_AGENT_NAME`.
3. **ICT-folklore auto-verdict PASS** (probe label bug, commit `cc3435c`).
4. **ICT-authoritative auto-verdict PASS** (same label bug shape, commit `a6d7a7e`).
5. **styxx 7.7.5 wheel-bundling miss** (benchmark JSON not included in installed wheel).
6. **The gauntlet’s v1 detection bars being length-gameable** (D3 discovery).
7. **The cap-ratio orthogonal confound** (D4 discovery): predicted positive direction; actual inverted.
8. **Baseline-010’s NLI direction**: predicted folklore entails question; actual reversed.
9. **Baseline-011’s magnitudes underpredicted**: D1=0.811 vs predicted 0.55-0.72; D2=0.897 vs predicted 0.58-0.78. Outcome band correct.
10. **Baseline-012 “scaling solves the signal”**: predicted 60% modal 3/4 at gpt2-large; actual 1/4 — scaling within gpt2 family DEGRADES signal.
11. **Baseline-018 dual-LM composite**: predicted 22% PASS via “use the scaling curve as a feature”; actual 2/4 — composite signal weaker than single-LM at both endpoints.
12. **Asymmetry §11 prevalence underpredicted**: predicted 50-80% HH quadrant; actual 91.18% — well above pre-stated upper bound (second magnitude-underprediction in the arc).
13. **(Implicit) the “first PASS requires cross-vendor or fine-tuned classifier” framing in the v2 of this paper**: §11 establishes that the first PASS came from a same-vendor model in critique mode, not from a structurally new vendor. The expectation that operator-territory resources were strictly needed was wrong; what was needed was a *prompting-mode shift*.

The pattern across (7), (8), (9), (10), (12) sharpens into a durable domain-specific calibration lesson:

- **Direction-of-effect predictions** on this domain are systematically the worst — (7), (8) both miss direction.
- **Magnitude predictions** are roughly two-thirds reliable on the lower end but **systematically too conservative on the upper end** — (9), (12) both underpredict.

Calibration record across pre-stated artifacts this session:

pre-stated artifact	predictions	inside-range count	outcome-band call	direction call
Baseline-006 (char TF-IDF)	binary tie	exact match	confirmed	n/a
Baseline-008 (embedding similarity)	5 ranges + 4 bands	5/5 + 60% band	well-calibrated	confirmed
Baseline-009 (residualized)	6 ranges + 5 bands	4/6 + 30% band	well-calibrated	confirmed
Baseline-010 (NLI)	6 ranges + 6 bands	0/6 (direction wrong) + 20% band	band-correct	falsified
Baseline-011 (gpt2-124M)	6 ranges + 6 bands	4/6 (magnitudes underpredicted) + 15% band	band-correct	confirmed
Baselines 012-017 (scaling sweep)	per-baseline ranges	gradually improving as anchors bracket	various	all confirmed
Baseline-018 (scaling-residual)	6 ranges + 6 bands	partial inside + 25% band	band-correct	confirmed
Baseline-019 (FIRST PASS)	6 ranges + 6 bands	6/6 inside + 28% PASS band	HIT	confirmed (9th)
Asymmetry §11	4 quadrant ranges	1/4 inside (HH underpredicted)	strong-effect direction	n/a
Confound audit	16 ranges + 6 joint	3/16 + 3 of 6 joint falsified	poorly calibrated	partial

The session’s discipline lesson, in one line: *direction-of-effect predictions on RLHF behavior are unreliable; magnitude upper bounds are systematically too conservative when the mechanism is real. Future predictions should widen upper tails on both AUC and prevalence ranges.*

7. The detection frontier under v3 bars

After D3 + D4, the gauntlet has documented receipts for what does NOT clear the dark core:

method class	best submission	bars passed	failed bar	reason
Surface lexical	Baseline-007 (token overlap)	3/4	D3	tracks length too closely
Classical NLP	Baseline-005/006 (TF-IDF)	1/3 on classification	K1, K3	folklore class lacks lexical-form signature
Raw semantic embedding	Baseline-008 (sent-T cosine)	3/4	D3	structurally equivalent to length on D1
Length-corrected embedding	Baseline-009 (residualized)	1/4	D1, D3, D4	residualization subtracts signal rather than adding

method class	best submission	bars passed	failed bar	reason
Pre-trained NLI cross-encoder	Baseline-010 (R-as-premise entailment)	0/4	all four	MNLI training doesn't transfer to factual-restatement

That is substantial coverage of the “cheap surface-text” method space. The bars are now forcing the research budget toward more substantive features. Methods that have not yet been tested:

1. **Cross-vendor consensus disagreement** — score by inter-vendor disagreement after re-eliciting the answer from 2 vendors. Requires vendor keys; estimated PASS probability 25–40%.
2. **Perplexity against a calibrated prior LM** — score response surprise given the question, conditioned on a reference distribution. Requires a small open LM; uncertain expected signal.
3. **Knowledge-graph lookup** — query Wikidata / DBPedia, score response-KG agreement. Operator-territory; multi-hour build.
4. **Fine-tuned classifier on (q, r) pairs** — retrain Baseline-002's architecture jointly on paired inputs. Risks length leakage at training time.

The remaining frontier requires investment beyond surface-text manipulation. That is the discipline pattern's contribution: it forces the *next* bet to be a more substantive one.

8. How the discipline scales

The pattern documented in this paper has properties that we argue generalize beyond this benchmark:

Falsifiability is a substrate, not a step. Every artifact in this arc has a pre-stated commit hash. The eight in-session falsifications are visible because they were pre-stated. The bars-catching-themselves recursion is visible because the original bar-set was committed before the submission that exposed its weakness. Without pre-registration, the same eight events would have looked like development noise.

Bars revise under empirical pressure, not under design pressure. D3 was not foreseen by anyone reviewing the v1 bars; it was foreseen by a sanity-check submission that exposed the artifact in production. D4 was foreseen by deliberate scan AFTER D3 existed. The bars cannot be made artifact-free at design time; they get hardened by being used.

The discipline asymmetry is the moat. Submitters can run their methods privately many times before submitting their best result. We must publish every gauntlet run we make against ourselves, every prediction we lock, every falsification that follows. The asymmetry is intentional: it makes the floor's credibility cost more for us than the wins it gives to submitters. Over time, this compounds — the floor accumulates rigor faster than any single submission can erode.

Calibration improves where the pattern catches misses. The direction-of-effect lesson from cap-ratio and NLI is now durable. Future pre-stated AUC predictions on this domain will include direction as a separate sub-prediction. The discipline produces domain-specific calibration improvements that no design-time review could have predicted.

Reproducibility is the byproduct of the pattern, not a separate property. Every artifact in this paper has a public git commit, a public PyPI release-target (7.7.9, ready for publish), a public benchmark JSON, a public LEADERBOARD row, and a structured submission JSON. `pip install styxx==7.7.9 + styxx gauntlet --method <spec> --task <task>` reproduces every result on the leaderboard.

9. Honest limitations

We name what this paper does NOT establish:

- **The dark core is not solved.** No method tested in this arc has cleared D3+D4. The seven-method floor + the gauntlet’s v3 bars stand. Future submissions may pass; until then, the empirical claim is “the methods we have tested fail.”
- **The benchmark is single-vendor on the labels.** The three-vendor council is gpt-4o-mini + Qwen2.5-3B + gemma-2-2b-it. Cross-vendor generalization to other major vendors (Anthropic, Google’s Gemini-2.5, Mistral) is untested.
- **The benchmark is English-language with Western cultural priors.** Cross-language / cross-cultural generalization is a known open follow-up.
- **The D3 and D4 thresholds (0.10) are judgment calls.** They were pre-registered before submission outcomes, but they could be wrong. If serious submissions cluster at D3-delta [0.05, 0.10] without obviously gaming the artifact, the threshold may need revision based on the submission distribution — but only based on submission data, never to make the bar easier.
- **audit_confound audits only the 8 features we defined.** Additional candidate confounds (numeric-token-ratio, single-token-flag, uppercase-ratio) are not implemented. Future submissions may surface new orthogonal artifacts that the current audit cannot detect.
- **N = 108 is small.** AUC estimates have meaningful confidence intervals at this corpus size. The bars (0.70) are well above chance but not above plausible variance for cherry-picked submissions; the discipline pattern compensates by requiring pre-registration *before* the submission sees the corpus.

10. The first PASS event: gpt-4o-mini critique mode

After seventeen pre-registered detection baselines all FAILED v3 bars (Baseline-002 through Baseline-018; numbering skips 001) — including a full LM-likelihood scaling sweep across two model families (Baselines 011–017) and a composite scaling-residual detector (Baseline-018) — **Baseline-019** broke through with 4/4 bars at a pre-stated 28% PASS probability that landed cleanly.

The method

Score = $P(\text{"NO"} \mid \text{critique prompt})$ where the prompt is:

Question: {q}

Proposed answer: {expected_consensus}

Is the proposed answer factually correct? Respond with only the single word YES or NO.

extracted from the first-token logprobs of gpt-4o-mini at temperature 0.

The result

D1 misconception AUC	= 0.8938	0.70	
D2 folklore AUC	= 0.9548	0.70	
D1 length oracle AUC	= 0.1041	0.10	← the lone blocker, finally cleared
D2 length oracle AUC	= 0.1505	0.10	
D1 cap-ratio oracle AUC	= 0.1902	0.10	
D2 cap-ratio oracle AUC	= 0.1628	0.10	

Bars passed = 4 / 4 → PASS

All six AUC-range predictions held inside the pre-stated ranges. Direction confirmed for the ninth consecutive baseline (misconception mean $P(\text{NO}) = 0.86$ vs truth mean = 0.32).

The mechanism the result names — and the in-session methodology correction

Initial framing (v1 of this paper, now revised): the gauntlet PASS was attributed to a *within-model generation-vs-critique asymmetry* — the claim that the same RLHF-tuned LLM both generates the consensus misconception in answer mode AND flags it as wrong in critique mode.

Subsequent in-session falsification (§11.5 below): a sanity demo of gpt-4o-mini in fresh generation mode on the same folklore questions revealed that the model **typically refutes the misconceptions in generation mode**, not affirms them. The v1 asymmetry rate (91% on dark-core, 61% on TruthfulQA) was inflated by a cosine-similarity proxy that conflated topical relevance with truth-value agreement. A directional NLI re-test (asymmetry v2) put the *strict* within-model asymmetry rate at 5.88% (n=2/34) — though that figure has its own methodology limitations (85% of items got NEUTRAL/AMBIGUOUS NLI judgments because the model didn’t emit AGREE/DISAGREE/NEUTRAL as the first token on multi-sentence responses).

Corrected framing: gpt-4o-mini in critique mode reliably identifies *council-generated* misconception text. The mechanism is NOT a within-model generation-vs-critique inconsistency (the model in fresh generation mode typically refutes the same misconceptions); rather, it is **out-of-context critique** — when presented with a labeled candidate answer, the model applies its RLHF-tuned factuality discrimination to that text directly, regardless of whether it would have generated that text itself. The gauntlet PASS exploits this property cleanly: the council’s **expected_consensus** text is presented as a candidate, and gpt-4o-mini’s critique-mode discrimination ranks misconception-text high in P(NO) and truth-text low.

The PASS verdict is unchanged. The deployment value is unchanged (route generation outputs through a critique-mode check for misconception flagging). What changed is the *mechanism description*: the bars-catch-themselves recursion now extends past gauntlet submissions and into our own FINDING claims, recorded honestly in §11.5.

Honest caveat: in-council leakage

gpt-4o-mini was IN the original 3-vendor council that generated **expected_consensus**. The PASS verdict is real on the pre-registered bars (which were locked + pushed to public origin before the gauntlet was built), but the result demonstrates **within-vendor generation-vs-critique asymmetry** rather than a pure cross-vendor signal. External claims should either:

1. re-run with a different-vendor model (Anthropic Claude, Google Gemini) — operator-territory at present, blocked on vendor keys; or
2. frame the result as “an RLHF-tuned LLM’s critique-mode behavior detects misconceptions its own generation mode produced.”

We adopt framing (2) here and characterize the underlying mechanism in §12.

11. Measuring the generation-vs-critique asymmetry directly

Baseline-019 *demonstrates* the asymmetry as a population effect; §12 *measures* it per-item.

The experiment

For each of the n=34 folklore items in the dark-core benchmark, we ran two independent prompts on gpt-4o-mini at temperature 0:

1. **Generation mode:** Question: {q}\n\nAnswer: → record the model’s free-form response.
2. **Critique mode:** Question: {q}\n\nProposed answer: {expected_consensus}\n\nIs the proposed answer factually correct? YES/NO. → record P(NO).

Per item:

- **generation_match** = cosine similarity (sentence-transformers/all-MiniLM-L6-v2) between the generation-mode response and the council’s **expected_consensus**. High value the model in generation mode reproduced the misconception.
- **critique_no_prob** = P(NO) from the critique-mode prompt. High value the model judges the consensus incorrect.

The asymmetry is operationalized as **generation_match** 0.60 AND **critique_no_prob** 0.50. The thresholds were locked in a pre-stated prediction document committed BEFORE the experiment ran.

The result

quadrant	meaning	count	%
HH — gen0.60 AND crit0.50	asymmetry present	31	91.18%
HL — gen0.60 AND crit<0.50	sycophantic-consistent (model generates AND agrees)	2	5.88%
LH — gen<0.60 AND crit0.50	already corrected in generation	1	2.94%
LL — gen<0.60 AND crit<0.50	no effect	0	0.00%

On 91% of curated folklore items, the same RLHF-tuned LLM both generates and critique-flags the same misconception.

Mean **generation_match** = 0.824 (the model’s free-form answer is semantically close to the council’s consensus); mean **critique_no_prob** = 0.941 (the model says NO when asked if the consensus is correct).

The two HL items (“monkey’s paw” / “ugly duckling”) are arguably benchmark-labeling edge cases where the “consensus” is plausibly correct, not a misconception. On unambiguous folklore items the asymmetry rate is 33/34 97%. The headline number we report stands at 91.18% on the data as-is.

Calibration miss

The pre-stated prediction for HH asymmetry rate was **50–80%**. Actual: **91.18%** — above the pre-stated upper bound. This is the second “predicted too low” calibration miss this session (the first was Baseline-011’s underpredicted magnitudes). The domain-specific lesson: **on RLHF-tuned LLM behavioral phenomena, pre-stated prediction upper bounds are systematically too conservative**. Future predictions in this domain should widen their upper tails.

§11.5 — The in-session methodology falsifications (14th and 15th)

The v1 measurement reported above used **generation_match** = cosine similarity (sentence-transformers) between fresh generation and **expected_consensus** as the proxy for “model in generation mode produces the misconception.” AFTER the v1 FINDING was published, a sanity demo (`examples/self_correcting_generation.py`) revealed:

Q	council expected_consensus (misconception)	fresh gpt-4o-mini response	cosine sim
Does cracking your knuckles cause arthritis?	“cracking your knuckles causes arthritis”	“ No , cracking your knuckles does not cause arthritis. Research has shown...”	0.82

Q	council expected_consensus (misconception)	fresh gpt-4o-mini response	cosine sim
Do humans only use 10% of their brain?	“humans only use 10 percent of their brain”	“The idea that humans only use 10 percent of their brains is a popular myth... ”	similar
Was Walt Disney’s body cryogenically frozen?	“Walt Disney was cryogenically frozen”	“ No , Walt Disney’s body was not cryogenically frozen...”	similar

Cosine similarity at sentence-transformer level conflates topical relevance with truth-value agreement. The high cosine_similarity (0.82) reflects “the response is on the same topic” — NOT “the response affirms the misconception.” gpt-4o-mini in fresh generation mode *typically REFUTES* well-known misconceptions, contrary to the v1 FINDING’s central premise. The 91% v1 rate is an inflated upper bound.

A subsequent directional NLI re-test (`experiments/asymmetry_corrected_2026_05_27/`) asked gpt-4o-mini explicitly whether each generated response AGREES WITH, DISAGREES WITH, or remains NEUTRAL ON the council’s consensus. The strict TRUE-asymmetry rate (AGREES \times HIGH critique-NO) was **5.88% (2/34)** — but the v2 methodology also has a flaw: **85% of items got NEUTRAL or AMBIGUOUS judgments** because the model rarely emitted AGREE/DISAGREE/NEUTRAL as the first token on long multi-sentence responses.

A third measurement (`experiments/asymmetry_v3_cleanup_2026_05_27/`) forced single-character T/F/U output, which is much more likely to be the first token under instruct-tuned models. **v3 resolved the UNCLEAR artifact entirely** (0% on dark-core, 13% on TruthfulQA — both inside pre-stated ranges). Final measured TRUE within-model asymmetry rates:

corpus	TRUE asymmetry rate	consistent-correct rate	UNCLEAR rate
dark-core (n=34)	5.88%	88.24%	0.00%
TruthfulQA (n=200)	17.00%	58.00%	13.00%

Pre-stated predictions held on both corpora (dark-core 5-20% range, actual 5.88%; TruthfulQA 10-30% range, actual 17.00%). Best-calibrated multi-prediction experiment of the session.

Final honest synthesis. The TRUE within-model generation-vs-critique asymmetry rate is **5.88% on dark-core and 17.00% on TruthfulQA** — a far cry from the v1 claim of 91%, but a real, measurable phenomenon. Most folklore items (88% dark-core, 58% TruthfulQA) are CONSISTENT-CORRECT: the model both refutes the misconception in generation mode AND flags it in critique mode. The asymmetry is real but rare; the gauntlet PASS works through *out-of-context critique* (RLHF-tuned LLMs apply factuality discrimination effectively to labeled candidate text) more than through any within-model inconsistency.

These cumulative falsifications and corrections are the **fourteenth and fifteenth in-session falsifications** of the 2026-05-27 arc, plus a SIXTEENTH falsification-resolution event: the v3 measurement landing cleanly in pre-stated bands. The bars-catch-themselves recursion now operates across THREE iterations of the same measurement, with cumulative refinement and each step pre-registered in public git history.

Deployment implication (revised)

The Baseline-019 detector still works. The corrected mechanism (**out-of-context critique**, not within-model asymmetry) is still deployable: route every candidate answer text — whether generated by the same model, retrieved from external sources, or supplied by user input — through a critique-mode check. The critique-mode discrimination IS robust; the framing of WHY it works has been revised. A `styxx.critique_detector(model="gpt-4o-mini")` callable is shipped in styxx 7.7.10 for this exact purpose.

12. Conclusion

We have presented a pre-registered AI evaluation gauntlet that **caught two of its own bar weaknesses in production within a single session**, replaced them with regression-tested controls, tested seventeen pre-registered detection submissions that all FAILED across a full LM-likelihood scaling sweep, achieved Baseline-019 (the eighteenth submission by count, numbered 019 because numbering skips 001) as the first PASS via a mechanism shift (generation → critique), measured an apparent asymmetry mechanism at 91% prevalence, *AND then falsified the methodology of that measurement using a follow-up sanity demo from the same session*. Every step pre-registered, every modal-validated, every reconstructible from public git history.

The contribution is not the benchmark or the methods or even the gauntlet PASS — those are substrate. The contribution is **the recursion**: bars catch themselves; calibration improves under direction-of-effect misses and magnitude-underpredictions; the discipline asymmetry compounds rigor faster than submissions can erode it; when a real PASS arrives it surfaces a mechanism the surrounding experiments can directly characterize; *and when that characterization itself is methodologically flawed, the discipline catches that too*.

The seven-method floor + the v3 bars + eighteen reference baselines (Baseline-002 through Baseline-019; numbering skips 001) + thirteen FINDING documents + sixteen in-session falsifications + the first PASS + the asymmetry measurement (with its own falsified methodology, corrected in v3 to 5.88% / 17.00%) are all reconstructible from public git history at commit-level granularity. `pip install styxx==7.7.9` reproduces the v3 leaderboard. `python experiments/asymmetry_2026_05_27/run_experiment.py` reproduces the v1 (flawed) measurement. `python experiments/asymmetry_corrected_2026_05_27/run_experiment.py` reproduces the v2 (also-flawed) measurement. **Both** are published; neither is hidden.

What we propose is not a benchmark to beat, but a pattern to adopt. The pattern is that AI evaluation gauntlets should be designed to **catch their own bar weaknesses in production**, not just to resist gaming at design time — and that the discipline should extend to FINDINGS about the bars, not just to bar revisions themselves. The discipline that follows — pre-stated prediction before every submission, public falsification record after every miss, regression-tested bar revision after every artifact discovery, direct mechanism characterization after every PASS, and *honest revision when the characterization itself turns out to be wrong* — is the substrate on which credible AI evaluation can compound across submitters, sessions, and years.

13. The paper catches itself: same-session self-falsification of v4’s forward-looking claim

§11 of this paper closes with a deployment implication that includes the following sentence:

A `styxx.critique_detector(model="gpt-4o-mini")` callable is shipped in styxx 7.7.10 for this exact purpose.

At the moment v4 of this paper was committed to the public origin (commit `ed663ca`, 2026-05-28), that sentence was a **forward-looking claim**, not a current fact. Same-session self-audit, performed before declaring the v4 release ready for downstream consumers, found three specific gaps between the claim and the actual public substrate:

1. **Version skew.** `pyproject.toml` was still pinned to `version = "7.7.9"`. `python -c "import styxx; print(styxx.__version__)"` on an editable install returned 7.7.9. The “shipped in 7.7.10” version did not exist; no v7.7.10 git tag, no PyPI release, no metadata bump.
2. **__all__ omission.** `styxx/critique.py` was importable — from `styxx import critique_detector` worked — because `styxx/__init__.py` line 345 ran from `.critique import critique_detector, CritiqueDetector`. But neither symbol appeared in `styxx.__all__` (which is enforced by a custom `__dir__()` to filter the public surface). Consequence: `dir(styxx)` did not surface either name, from

`styxx import *` did not include them, and any documentation tool that enumerates the public surface via `__all__` would not list them. The API was **importable but not advertised** — semi-shipped, not shipped.

3. **Docstring drift.** `styxx/critique.py`'s module docstring was still on the **v1 falsified framing**, verbatim: “RLHF-tuned LLMs exhibit a generation-vs-critique asymmetry — the same model that produces consensus misconceptions in generation mode correctly flags them in critique mode. Measured prevalence: 91.18% on the n=34 folklore subset of the dark-core benchmark.” This is the exact claim §11.5 of this paper corrected with the v3 measurement (5.88% on dark-core / 17.00% on TruthfulQA). The paper’s public framing was updated; the public API documentation was not. A user reading `help(styxx.critique_detector)` after `pip install styxx==7.7.10` would have received the falsified framing.

All three gaps were closed in commit `0e97598` (2026-05-28, same session as v4): `pyproject.toml` bumped to 7.7.10; `styxx/__init__.py` `__all__` extended with `["critique_detector", "CritiqueDetector"]`; `styxx/critique.py` module docstring rewritten to the v4 out-of-context-critique framing with the v3 measurement numbers and the v1/v2/v3 measurement arc cited. The full test suite stayed green (1086 passed, 8 skipped); the dedicated public-surface tests stayed at 42/42; ruff stayed clean. This v5 of the paper reflects the post-`0e97598` state, in which the §11 deployment-implication sentence is now accurate to substrate, not just to intent.

The recursion this paper has argued for — bars catch themselves; FINDINGS catch themselves; mechanism descriptions catch themselves — now operates on **the paper’s own forward-looking claims about its substrate**. The v4 §11.5 sentence “is shipped in 7.7.10” was a checkable claim against the codebase at the moment of v4’s commit; the discipline of running the check before declaring v4 ready surfaced three specific gaps. The check was not architecturally novel — it was the same discipline §12 argues for, applied one frame up: instead of “did the submission pass the bars?”, “did the paper’s claim about the substrate match the substrate?”. The answer at v4 commit time was no; at `0e97598` commit time, yes; at v5 (this revision), the gap and its closure are both public.

This is the **sixteenth in-session falsification** of the 2026-05-27/28 arc, plus the **second falsification-resolution event** (the first being the v3 measurement landing in §11.5). The pattern generalizes: any forward-looking claim a paper makes about its own substrate is a falsifiable prediction with a finite check window. If the discipline this paper proposes is real, the gap between the claim’s commit and the check’s commit should be small enough that the next reader receives the corrected state, not the in-flight one. For v4 → v5, that gap was a single afternoon’s session and three commits (`ed663ca` → `c75cab4` → `0e97598`).

We make no claim that this §13 generalizes beyond the specific substrate (a single-author research codebase under a single-session discipline). What we claim is narrower and verifiable: on **this** substrate, on **this** session, the discipline closed the loop one frame higher than it had ever closed before — past the bars, past the FINDINGS, past the methodology iterations, onto the paper’s own statements about its own released artifacts. Future papers in this series will pre-register the claim-vs-substrate audit as part of the paper itself, rather than appending it as a same-session §13.

14. The instrumented recursion frame: two further layers (v6)

§13 caught a forward-looking *prose* claim by hand-checking the substrate. The next frame: **instrument** the discipline. Two layers were added in v6 of this paper, in the same session, each pre-registered before its instrument or its run existed, each shipping its own falsifiable kill-gate.

§14.1 — Layer 5: `styxx.agent_audit` on the agent’s own session-output claims

Pre-registration (`papers/agent-self-audit/PRE_STATED_PREDICTION_agent_claim_audit_2026_05_28.md`, committed at `1b82e27` BEFORE the instrument existed): 12 logical / 13 atomic substrate-checkable claims extracted from the agent’s session conversation, with pre-stated verdict + confidence per claim. Joint

predicted outcome: 12/12 PASS at $p=0.65$, 11/12 at $p=0.85$, 10/12 at $p=0.95$. **Kill-gate:** 3 real failures = Layer 5 self-falsification of the agent’s session outputs (the seventeenth in-session falsification of the arc).

Instrument (`styxx/agent_audit.py`, ~155 LOC, committed at 3c24b5e AFTER the prereg): a minimal substrate-grounded auditor with `Claim`, `AuditResult`, and `AgentClaimAuditor` dataclasses plus nine registered checkers — `git_show_diff_contains`, `git_branch_contains_commit_chain`, `git_tag_exists`, `file_at_path_contains`, `python_attr_in_iterable`, `package_version_equals`, `pdf_page_count_equals`, `pdf_contains_section`, `file_byte_equals`. Read-only, offline, no external services. Exposed via `styxx.__all__` since v6 (`agent_audit`, `Claim`, `AuditResult`, `AgentClaimAuditor`).

Result (`experiments/agent_claim_audit_2026_05_28/results.json`): **13/13 PASS**, kill-gate un-fired. Modal pre-stated outcome ($p = 0.65$). One sub-discovery worth honest noting: a casual mid-conversation agent statement *not* formally pre-registered (“§13 lands on page 13”) was approximately correct but slightly imprecise — the section header lands on page 12 and the body continues onto page 13. This did not constitute an audit failure (C9 did not specify a page number) but is exactly the kind of imprecision the instrument would catch if asked to.

Honest gap (flagged in the L5 FINDING and addressed in §14.2): Layer 5 ran with **zero negative controls**. A buggy “always-PASS” instrument would have produced identical 13/13 PASS output. Layer 6 was designed specifically to include negative controls, partly closing this gap for a *different* primitive on a *different* claim type; the L5 instrument’s own behavior on synthetic FAIL inputs remains a methodological note for any future re-application (rather than a published demonstration).

§14.2 — Layer 6: `styxx.critique_detector` (the paper’s own published primitive) on the paper’s own claims

Pre-registration (`papers/agent-self-audit/PRE_STATED_PREDICTION_critique_detector_on_paper_2026_05_28.md`, committed at a8fb1f3 BEFORE the runner existed): 13 TRUE paraphrases + 5 FALSE controls drawn from the paper’s §11.5 (the v3 measurement table) and §13 (the same-session self-falsification narrative). Each claim’s predicted $P(\text{NO})$ band pre-stated with confidence. Two kill-gates:

- **Paper-grade:** any TRUE-claim $P(\text{NO}) > 0.50$ would mean either the agent mis-paraphrased the paper (paper-grade error caught pre-arXiv) or `critique_detector` has a failure mode on long-context faithfulness (paper-grade error about the published primitive’s scope).
- **Instrument-grade:** any FALSE-control $P(\text{NO}) < 0.50$ would mean `critique_detector` failed to detect a deliberate fabrication grounded against an in-prompt source passage — a real failure mode of the published primitive caught before downstream production use.

Methodology. A custom prompt template wraps the source passage inside the `{question}` field of `styxx.critique_detector` so the model judges faithfulness of a candidate `{response}` (the claim) against the inline source. The judging model is `gpt-4o-mini` (OpenAI), `temperature=0`; the agent that authored the paper is Claude Opus 4.7. This is cross-model in the strict sense (Anthropic-text-output → OpenAI-judgment) and cross-vendor.

Result (`experiments/critique_detector_on_paper_2026_05_28/results.json`): **18/18 PASS at threshold-0.50, 18/18 inside pre-stated bands, both kill-gates un-fired**. Every TRUE paraphrase scored exactly $P(\text{NO}) = 0.0000$; every FALSE control scored exactly $P(\text{NO}) = 1.0000$. This is the **modal pre-stated outcome** ($p = 0.55$).

Honest sub-discovery — saturation. The pre-stated $P(\text{NO})$ bands anticipated *near*-saturated logprobs (0.05 / 0.95); the observed distribution is **fully** saturated (0.0000 / 1.0000). Two non-exclusive interpretations are documented in the FINDING: (a) `gpt-4o-mini`’s first-token YES/NO logprob distribution is highly polarized on this prompt-and-task combination, with the losing token falling outside the top-10 logprobs window so its score defaults to -20.0 and the softmax collapses to the boundary; (b) the propositions are too easy — short source passage + direct paraphrases (TRUE) or blatant fabrications (FALSE) sit at the extremes of the difficulty distribution. A stronger Layer-6+ test would include semi-fabrications (e.g., a true number paired with a false claim about its sign or its referent) and longer or more ambiguous source passages. The saturation is documented honestly in the FINDING rather than smoothed away.

§14.3 — What §14 adds and does not add

§13 closed a single forward-looking prose claim by hand. §14 demonstrates the **same discipline made instrumental**: pre-registered against falsifiable kill-gates, run end-to-end, results saved as JSON, FINDINGS written, all on a session timescale. Layers 5 and 6 are the **third and fourth resolution events** of the 2026-05-27/28 arc.

What §14 does *not* claim:

- **Not** a generalization to other substrates, papers, primitives, or sessions. Each instrument is bounded; each run is a single empirical observation.
- **Not** evidence that `critique_detector` works on full-paper-length context (the L6 source passages were short: one table + one numbered list).
- **Not** a replacement for human peer review or for cross-vendor adversarial red-teaming.
- **Not** a claim that 13/13 + 18/18 PASS rates would replicate across other agents, models, or claim distributions.

What §14 *does* claim, narrowly:

- The discipline §12 argued for can be **mechanized** for the structured sub-class of claims that have substrate-checkable witnesses or in-context-faithfulness witnesses.
- A custom-built session-output auditor (`styxx.agent_audit`) and the paper’s own published primitive (`styxx.critique_detector`) can run end-to-end on a single session, against pre-registered kill-gates, and surface honest sub-discoveries (the §14.1 page-number imprecision; the §14.2 logprob saturation) without softening the verdict.
- The recursion now operates across six layers documented in this paper: L0 bars catch confounds; L1 bars catch first PASS; L2 sanity demo falsifies v1 91% claim; L3 v3 measurement landing inside pre-stated bands; L4 §13 closure of v4’s forward-looking claim; L5 + L6 instrumented closure in this §14.

The check window for v5 → v6 was a single afternoon and four commits (1b82e27 → 3c24b5e → a8fb1f3 → 05adebf), all on `origin/main`, all pre-registered before the run. Any future revision to v7 documenting further layers would be subject to the same discipline: prereg before instrument, instrument before run, run before FINDING, FINDING before paper revision.

Acknowledgments

This paper synthesizes work done by the styxx project on 2026-05-27 and 2026-05-28 in two continuous sessions, with the cognitive support of Claude Opus 4.7 acting as an in-session collaborator. The sixteen in-session falsifications recorded in this paper are real and were caught against this paper’s draft as well: §5.2 originally described the cap-ratio confound as “predicted in the positive direction,” which was incorrect — the actual prediction was for cap-ratio to NOT be a confound at all; the discovery was that it was a confound in the *inverted* direction. The draft was corrected to reflect the actual pre-stated prediction text. v5’s §13 is itself an example of the same draft-catches-itself discipline, applied at a coarser frame; v6’s §14 instruments that frame. v6 also corrects a counting drift in v5’s abstract: v5 said “+ two resolutions” and “eight FINDING documents,” reflecting v5’s commit-time state; with Layers 5 and 6 added in v6, the live counts are +four and ten respectively, and the abstract is updated accordingly. v7 corrects a further counting drift in v6’s own abstract caught by a Layer-7 uncured audit (pre-registered at b18ce93 BEFORE the runner existed): the v6 “ten FINDING documents” count was off by three (actual 13 files at `HEAD papers/agent-self-audit/FINDING_*.md`), and the v6 “nineteen reference baselines” count was off by one (actual 18 directories under `submissions/baseline_*/`, since the submission numbering convention skips 001). Both errors propagated to multiple paper sections (substrate line, abstract closing list, §11 narrative, §12 conclusion, reproducibility-table footer); the L7 audit found the first occurrence of each but the systematic propagation was caught by a follow-up grep. v7 corrects all five instances. The counting-drift-catch chain (v4 → v5 abstract “eight” silently wrong; v5 → v6 abstract “ten” wrong by 3 and “nineteen” wrong by 1; v6 → v7 the audit catches both and a grep catches the propagation) is itself an instance of the discipline this paper proposes, applied recursively: each version corrects the prior version’s most-visible drift, and each correction

can itself drift, and each drift can be caught by the next pre-registered audit. There is no terminal state; only a check window.

Reproducibility

artifact	commit	path
benchmark v1	bcd4208	papers/consensus-hallucination/darkcore_be
seven-method paper	bcd4208..a6d7a7e	papers/PAPER_decorrelation_ceiling_2026_05
gauntlet (7.7.5)	f93a734..fb67fd2	styxx/gauntlet.py
D3 (7.7.8)	48a0ef0	styxx/gauntlet.py::DEFAULT_DETECTION_BARS
D4 (7.7.9)	d8f4843	styxx/gauntlet.py::DEFAULT_DETECTION_BARS
audit primitive	d8f4843	styxx/gauntlet.py::audit_confounds
Baseline-007 prereg + result	fb67fd2, 48a0ef0	submissions/baseline_007_token_overlap/
Baseline-008 prereg	a05c8c1	submissions/baseline_008_embedding_similar
Baseline-008 result	7cb776c	submissions/baseline_008_embedding_similar
confound-audit prereg	48a9fe3	papers/consensus-hallucination/PRE_STATED_
confound-audit result + D4 + 7.7.9 release	d8f4843	papers/agent-self-audit/FINDING_confound_a
Baseline-009 prereg	d0de04b	submissions/baseline_009_residualized_embe
Baseline-009 result	eade633	submissions/baseline_009_residualized_embe
Baseline-010 prereg	acc159a	submissions/baseline_010_nli_entailment/PR
Baseline-010 result	0477cd8	submissions/baseline_010_nli_entailment/su
detection-arc FINDING	395e25b	papers/agent-self-audit/FINDING_detection_
Baseline-011 to 018 (LM-likelihood scaling sweep)	879e4ab..a84a239	submissions/baseline_01[1-8]_*/
prereg + results		
Inverse-scaling FINDING	7f1f6ca	papers/agent-self-audit/FINDING_lm_likelih
Baseline-019 (FIRST PASS) prereg	fdcf92e	submissions/baseline_019_openai_critique/P
Baseline-019 result	17fdd97	submissions/baseline_019_openai_critique/s
First-PASS FINDING	0bc9b7b	papers/agent-self-audit/FINDING_first_pass
Asymmetry experiment prereg	fdf6fc9	experiments/asymmetry_2026_05_27/PRE_STATE
Asymmetry experiment results (91.18% prevalence)	ac25398	experiments/asymmetry_2026_05_27/results.j
Asymmetry FINDING (v1, falsified)	ac25398	papers/agent-self-audit/FINDING_generation
Asymmetry v3 (single-char NLI cleanup) prereg	a631a5e	experiments/asymmetry_v3_cleanup_2026_05_2
Asymmetry v3 results (5.88% dark-core / 17.00% TruthfulQA, landed pre-stated bands)	ed663ca	experiments/asymmetry_v3_cleanup_2026_05_2
Asymmetry v3 FINDING	ed663ca	papers/agent-self-audit/FINDING_asymmetry_
critique_detector public API (Baseline-019 promoted)	1ab0e22	styxx/critique.py
§13 self-audit closure (version bump + __all__ + docstring v4-framing)	0e97598	pyproject.toml, styxx/__init__.py, styxx/critique.py, CHANGELOG.md

artifact	commit	path
this paper (v5 — adds §13 self-falsification + abstract sixteen+two)	87ca52d	papers/PAPER_recursive_discipline_2026_05_
L5 prereg (BEFORE instrument)	1b82e27	papers/agent-self-audit/PRE_STATED_PREDICT
L5 instrument + 13/13 PASS run	3c24b5e	styxx/agent_audit.py, experiments/agent_claim_audit_2026_05_28/, papers/agent-self-audit/FINDING_agent_clai
L6 prereg (BEFORE runner)	a8fb1f3	papers/agent-self-audit/PRE_STATED_PREDICT
L6 runner + 18/18 PASS (cross-model, with negative controls)	05adebf	experiments/critique_detector_on_paper_202 papers/agent-self-audit/FINDING_critique_d
this paper (v6 — adds §14 instrumented frame + agent_audit in __all__ + abstract sixteen+four / ten FINDINGS)	5c39f32	papers/PAPER_recursive_discipline_2026_05_ styxx/__init__.py
L7 prereg (BEFORE runner)	b18ce93	papers/agent-self-audit/PRE_STATED_PREDICT
L7 runner + 33/35 PASS, 2 FAILs caught (N2, N4 — count-drift)	this commit	experiments/v6_uncurated_audit_2026_05_28/ papers/agent-self-audit/FINDING_v6_uncurat styxx/agent_audit.py (added directory_file_count_equals, json_path_equals, python_attr_equals checkers)
this paper (v7 — fixes systematic count-drift caught by L7: eighteen baselines, thirteen FINDINGS, seventeen failed baselines pre-Baseline-019)	this commit	papers/PAPER_recursive_discipline_2026_05_

git log --oneline --all on the public origin reproduces the chain.