Failure Modes and Falsification Standards for Universal Collapse Theory

Where the Program Can Fail — Per Claim, Per Layer, and as a Whole

Standards Reference · read alongside the *Program Map*

Jeremy C. Jones
HoldingLight LLC — ORCID 0009-0007-2515-3774 — universalcollapse.com
Version: v1.0 (2026-06) | License: CC BY 4.0
DOI: 10.17605/OSF.IO/TN7Z3

What this document is

A program that asks its readers to look for structure has a standing obligation: to say, precisely, where its claims can fail. Without that, "look for structure" reads as unfalsifiable — a vocabulary that can absorb any outcome. This document discharges that obligation for Universal Collapse Theory (UCT). It is the consolidated falsification standard: a single place that gathers where each claim fails, where each layer fails, and how the program as a whole is judged over time.

These conditions are distributed across the corpus; this document consolidates them into one reference. Each is already written into the individual papers — each Technical Note states an acceptance and a rejection criterion, each Methods paper is built around a sharp falsifier, each empirical demonstration carries pre-specified falsifier conditions, and *The Structuralization of Empiricism* states five divergence claims with explicit failure conditions. This document collects that distributed material into one canonical reference, adds the per-layer and per-Prime failure modes, and states the program-level standard in full. The *Program Map* identifies where falsification enters the architecture; this document is where those conditions are stated completely.

> **What this document does not claim**
>
> This document does not prove that UCT survives any of these tests. It states the conditions under which it would fail. A program is made falsifiable by naming such conditions in advance and reporting outcomes against them honestly — not by passing them. Where the corpus records a failure against one of these conditions, that failure is part of the standard's function, not a defect in it.

1. The structure of falsification in UCT

Falsification in UCT operates at three levels, and a reader should know which level a given condition belongs to, because they fail differently and are judged differently.

- **Per-claim.** Individual discriminators — the signature bounds, their audits, the Prime reporting standards — each carry a specific observation that would count against them. This is where UCT can be wrong in the ordinary scientific sense.

- **Integrity architecture.** The standards layer names the ways a record-based framework can self-seal — corrupted records, pseudo-redundancy, frozen constraints, selective update — and the conditions under which those failures are detected. A framework that could not fail this way could not be trusted when it succeeds.
- **Program-level.** The program as a whole is not refuted by any single result. It is judged, longitudinally, as progressive or degenerating — a verdict read off an accumulating track record, not a stated threshold.

One distinction governs the first level. UCT separates three claim grades. **Level 1** is the formal kernel — it fails on grounds of internal incoherence or definitional inadequacy, not by experiment. **Level 2** is the structural-interpretive reading — it fails if a rival reading explains the same corpus with fewer commitments, or if it adds no explanatory gain. **Level 3** is the domain discriminators — the signature claims and their falsifiers — and this is the empirically testable portion of the framework. The sections that follow concern Level 3 primarily, with the integrity and program-level standards governing the whole.

1.1 Claim status vocabulary

The standard uses a fixed set of outcome states, defined here so they are read the same way throughout. These statuses are first-class: an inconclusive, inaccessible, or null result is a real outcome, not a missing one.

Status	Meaning
Supported	The prediction or discriminator appears under the stated conditions.
Not supported / absent	The predicted effect does not appear under the stated conditions.
Failed locally	A claim's falsifier fires within its declared domain.
Inconclusive	The test does not adjudicate the claim, because data, power, access, or assumptions are insufficient.
Inaccessible	A required layer cannot be observed under the access tier in force.
Bounded null	No effect within stated bounds — informative when the bound is specified.
Confounded	The audit cannot separate the target signature from an alternative explanation.
Global challenge	Systematic local failures across domains that independently satisfy the preconditions.
Standards failure	The integrity architecture itself fails: incomplete ledger, unrecorded failures, selective updates.
Degeneration	A ledger-level pattern of ad hoc rescue, narrowing, relabeling, or test removal over time.

One boundary matters most. A *not-supported or absent* result becomes a *local failure* only when the declared access, assumptions, power, and falsifier conditions were satisfied. Where they were not, the honest report is *inconclusive, inaccessible, bounded null, or confounded*, as appropriate. This guard runs both ways: it keeps a missing effect from being overclaimed as a refutation, and keeps a genuine failure from being downgraded into ambiguity.

2. Signature-level falsification (S_1, S_2, S_3)

Each structural signature is defended by a formal bound, operationalized by an audit, and exposed to data. Each layer carries its own failure condition, and they compound: a signature claim is in good standing only if the bound holds under its assumptions, the audit can separate the signal from confounds, and the empirical run does not come out against it. The table states all three failure conditions per signature.

Signature	Bound fails if (Technical Note)	Audit / empirical fails if (Methods, demonstration)
S_1 redundancy → consensus	The exponential consensus bound is mathematically incorrect, or the independence and discriminability assumptions are misstated.	The agreement curve fails the predicted (Chernoff-rate) decay under verified independence, or correlated pseudo-redundancy accounts for the observed consensus.
S_2 neutrality → delay	The first-passage analysis is incorrect, or the monotonic relationship between bias magnitude and resolution time fails under the stated drift-diffusion model.	The latency curve does not peak near neutrality, or departs from the predicted $E[\tau]$ shape under the stated model, a valid independent ΔK, and stated confound checks.
S_3 constraint sweep → hysteresis	The loop-area lemma $A(R) = 4\theta_0 + 4\alpha R$ is incorrect under its single-hysteron, quasi-static, major-loop assumptions, or audited record state cannot serve as a covariate.	The quasi-static loop area fails the predicted R-dependent scaling under a valid sweep, or the path dependence is fully explained by rate lag.

3. The framework's own divergence claims

Beyond the per-signature falsifiers, *The Structuralization of Empiricism* states five Level-3 divergence claims: narrow, testable predictions of what should be observed if the structural model is correct, each with a stated condition under which the framework must be revised or constrained. Four define local failure conditions; the fifth is a global portability challenge. These are the framework's self-stated falsifiers, reproduced here as the canonical list.

#	Divergence claim	Falsifier — the framework fails locally if
DC1	**Redundancy as a convergence driver.** Verified independent redundancy predicts measurable reduction in inter-observer disagreement.	Under verified independence and increasing redundancy, no measurable reduction in disagreement appears (S_1 fails locally).
DC2	**Neutrality-induced latency.** Under systematically reduced prior constraint, resolution latency increases.	Controlled prior-neutrality manipulation does not increase latency, or reverses the predicted latency relation, under valid audit conditions (S_2 fails locally).
DC3	**Record-dependent hysteresis.** Systems that accumulate records show path dependence under bidirectional constraint sweeps.	Systems with accumulated records show no discrepancy under controlled bidirectional sweeps (S_3 fails locally).
DC4	**Integrity interventions alter stabilization.** Independently improving or degrading records, update, or independence should alter stabilization in the predicted direction.	No change in stabilization appears across repeated controlled comparisons (the update-integrity claim fails locally).

#	Divergence claim	Falsifier — the framework fails locally if
DC5	**Cross-domain portability.** The S_1–S_3 family applies across all domains satisfying the recursive preconditions. (Global challenge.)	A domain independently satisfies the stated recursive preconditions — records, constraints, update loop, and a measurable signature channel — yet the relevant S-signatures systematically fail or collapse into confounds across repeated tests (the portability claim is challenged).

4. Hygiene-layer failure: the Prime reporting standards

The Prime series is conceptual hygiene, not empirical claim — but each Prime still defines a failure condition for its reporting standard, the point at which a claim made under that standard does not hold. These are the conditions under which a coherence-first analysis of the relevant term is wrong on its own terms.

Prime	A claim under this standard fails when
P0 Coherence	No constraint-visible structure survives testing — the claimed invariant or statistic does not hold under the stated audit, so there is nothing for "coherence" to name.
P1 Randomness	A scoped residual label is promoted into an ontic claim without exhausting the declared search (model class M, constraints K, dataset D, budget B), or discovered structure within the stated scope is ignored and the residual continues to be treated as primitive randomness. Overturning a provisional label when better structure appears is the success case, not the failure.
P2 Chaos	No persistent invariant or statistical structure is demonstrated under a valid test, so the chaos claim is downgraded to noise, stochastic forcing, unknown mixture, or unsupported; or the reported precision horizon and data-adequacy conditions of the Chaos Reporting Standard are not met.
P3 Intelligence	Plastic update cannot be causally tied to transfer gains. A positive adaptive-gain metric without demonstrated transfer is adaptive gain, not intelligence; the claim fails if the update loop, transfer, and boundary are not jointly shown.
P4 Nothingness	A "from nothing" or ground claim cannot name its ground, its rule, and a discriminator. If it can, it is not "nothing"; if it cannot, it is non-operational. Either way the absence-as-ground claim fails.

5. Per-layer failure modes

Each architectural layer of the program has a characteristic way of failing. A layer is in good standing to the extent that its characteristic failure has been looked for under appropriate conditions and not found, or has been found and repaired transparently. This table is the canonical statement of the per-layer failure modes summarized in the Program Map.

Layer	What failure looks like
Formal kernel	Internal inconsistency, ambiguous primitives, or definitions that do not port across domains.

Layer	What failure looks like
Structural interpretation	A rival reading explains the same corpus with fewer commitments, or the interpretation adds no explanatory gain over plainer alternatives.
Domain mapping	No discriminator emerges; the mapping improves no compression, prediction, transfer, or control over alternatives, and so remains metaphor.
Technical Note	The bound is mathematically incorrect, or fails under its own stated assumptions.
Methods paper	The audit cannot separate the target signature from confounds, or its falsifier cannot be made sharp under stated conditions.
Empirical demonstration	The pre-specified falsifier fires, the predicted effect is absent, or a positive result fails replication.
Standards layer	The ledger is incomplete, failures go unrecorded, or update rules are applied selectively — the framework self-seals.
Bridges	A reader-facing analogy distorts rather than clarifies the formal claim it is meant to introduce.
Applied / commercial	A diagnostic use fails external audit, or an applied result is mistaken for — or presented as — evidence for the law-level program.

6. The integrity architecture: failure of the update loop

A record-based framework has a distinctive way of going wrong: not by being refuted, but by *self-sealing* — quietly corrupting the loop by which records revise constraints, so that the framework appears to converge while only confirming itself. The standards layer exists to make those failures detectable. The *Update Integrity Standard* names the failure modes of the update loop; for each, a system that exhibits it has compromised update integrity to some degree, and a repair protocol is specified.

Corruptor	The update loop fails by
Record falsification	Altering or fabricating the durable trace, so later constraints update against a false record.
Pseudo-redundancy	Treating correlated trials of a shared assumption as independent confirmation — the appearance of convergence without the substance.
Constraint freezing	Refusing to let records revise constraints that they should revise; the update map U stops responding to disconfirming evidence.
Selective update	Applying revision rules only to favorable outcomes, so the constraint set drifts toward confirmation.
Coercive agreement	Manufacturing consensus by pressure rather than by independent records.
Discriminator-free coherence	Achieving internal consistency with no observation that could have come out otherwise — coherence without contact.
Identity binding	Fusing the framework's coherence to the holder's identity, so that revising the framework is felt as a threat to the self and is resisted for that reason.

Severity and locality. A corruptor can be local, systemic, repaired, or unresolved. Its presence is not automatically a standards-level failure: the failure state becomes standards-level when the corruptor materially affects the ledger, the update map, or the independence of records *and* is not transparently repaired. A detected-and-repaired corruptor, recorded as such, is the standards layer working, not failing.

Two standards-layer documents carry the framework's own exposure to failure. *Records Across Nature, Life, and Mind* makes the persistence layer first-class, so that record corruption and loss are nameable failures rather than silent ones. *The Structuralization of Empiricism* states the five divergence claims of Section 3 and the conditions distinguishing a stable empirical regime from a self-sealing one. Together they are the machinery that keeps the framework corrigible — and a standards layer whose ledger is incomplete or whose corruptor checks are not run is itself in the failure state named in Section 5.

7. Program-level falsification

The program as a whole is *not* falsified by any single result, and it does not carry a stated threshold whose breach ends it. That would be a category error. A research "programme" in Lakatos's sense is not refuted by a failed prediction but is judged, over time, as *progressive* or *degenerating*: progressive when it keeps making new domains structurally legible and yielding fresh discriminators; degenerating when it survives only by relabeling what it cannot predict and bolting on exceptions to save its core. That verdict is empirical and longitudinal, read off an accumulating track record — not a philosophical proof, and not reducible to a single threshold stated in advance.

This places one obligation on the program that *can* be discharged now: honest bookkeeping of every audit outcome, so the verdict is eventually read off a complete ledger rather than a curated subset. Selectively remembering successes — reporting the favorable results and quietly dropping the null and failed ones — would make "progressive" a curated success narrative rather than a finding. The guard is the standing commitment to record failures with the wins. The corpus already does this: the COGITATE reanalysis publishes a failed prediction (B, $p = 0.748$) and an inconclusive one (C) beside its supported one; the S3-RAG-01 study publishes a bounded null. The Update Integrity Standard's Empirical Ledger is the instrument that keeps the full account.

To keep that judgment from staying vague, the ledger is reviewed at release milestones — each review recording new discriminators, confirmations, nulls, failures, assumption changes, and any ad hoc repairs made to preserve a claim. The program reads as *progressive* when new discriminators and successful cross-domain transfers accumulate without a rising exception load, and as *degenerating* when negative outcomes are absorbed only by narrowing assumptions, relabeling results, or removing tests from view. This is a commitment to visible accounting, not a single-failure threshold: it makes the longitudinal verdict legible without pretending it can be triggered by any one result.

7.1 Minimum ledger fields

To make the milestone review concrete rather than aspirational, each ledger entry records at least the following. The schema is the operational form of the commitment above, and aligns with the Update Integrity Standard's Empirical Ledger.

Field	Purpose
Claim / artifact	What is being tested.
Pre-specified falsifier	What would count against it, stated before the test.
Dataset / domain	Where it was tested.
Outcome	One or more claim-status values from §1.1 — e.g. supported, not supported / absent, failed locally, inconclusive, inaccessible, bounded null, or confounded.
Assumption changes	What, if anything, changed after the result.
Repair status	None / proposed / implemented / validated.
Exception load	Whether a new exception was added to preserve a claim.
Replication status	Unreplicated / replicated / failed replication.

7.2 Capstone law status (pending)

This standard is silent on one failure condition by design. **WP05**, the capstone, is held for last and will state the law-level drift test for the candidate Law of Coherence. Until that paper is deposited, this standard treats law-level failure as *pending specification*. The existing Level-3 signature failures, the divergence-claim falsifiers, and the integrity conditions constrain the program *locally*; they do not by themselves adjudicate the capstone law unless WP05 defines that relation. A complete falsification standard for UCT therefore awaits WP05 — and this gap is marked, not hidden, which is the posture the rest of this document takes toward every condition it states.

8. What would count against UCT

Consolidated, the conditions under which the program fails fall into three kinds. Local failure is read off the individual papers; standards failure is read off the integrity architecture; program-level degeneration is read off the full ledger over time.

Local failure (claim, layer, or hygiene standard)

A signature bound fails under its own stated assumptions; an audit's sharp falsifier fires; a pre-specified empirical prediction is absent or fails replication; a divergence claim's falsifier is met.

A domain mapping yields no discriminator — it improves no compression, prediction, transfer, or control over alternatives, and remains metaphor.

A hygiene standard is violated — randomness promoted to an ontic claim without exhausting declared search; an intelligence claim without demonstrated transfer; a ground claim that cannot name ground, rule, and discriminator; a coherence claim whose invariant does not survive testing.

Standards failure (the integrity architecture itself)

> The ledger is incomplete, failures go unrecorded, update rules are applied selectively, or applied / commercial results are presented as evidence for the law-level program without public audit and ledger entry. This is the failure mode distinctive to a record-based framework: not being wrong, but ceasing to be corrigible.

> **Program-level degeneration (read off the full ledger)**
>
> A sustained pattern in which new mappings stop producing records, discriminators, or updateable results — in which the kernel is preserved only by ad hoc exception rather than by making new domains legible, and in which the exception load rises while failures are absorbed by narrowing assumptions, relabeling, or removing tests from view. No single failure triggers this; the degenerating trend, read off a complete and honestly kept ledger, is the signal.

None of these conditions has been waived, and the program does not claim to have passed them — it claims to have stated them, exposed its claims to them, and recorded the outcomes, including the failures, where the tests have been run. That is what makes the standard a standard rather than a defense: it would register a loss if one occurred; where losses have occurred, they have been registered.

9. Source artifacts

Every condition in this standard is consolidated from a source artifact. The table maps each to its locus, so the standard is auditable against the papers it summarizes.

Source artifact	Role in this standard	Identifier / status
Kernel First — Collapse Without Reification	Level 1 / Level 2 claim grades; anti-reification reading (§1)	10.17605/OSF.IO/6RZQ2
TN-S_1 — Objectivity from Records	S_1 formal bound and rejection criterion	10.17605/OSF.IO/6M7N3
TN-S_2 — Neutrality Delays Resolution	S_2 formal bound and rejection criterion	10.17605/OSF.IO/6WRQV
TN-S_3 — Records Amplify Hysteresis	S_3 formal bound and rejection criterion	10.17605/OSF.IO/QJMSZ
Methods-S_1 — Auditing Independence	S_1 audit falsifier	10.17605/OSF.IO/7U8SK
Methods-S_2 — Auditing Constraint Asymmetry	S_2 audit falsifier	10.17605/OSF.IO/HRKWT
Methods-S_3 — Auditing Record State	S_3 audit falsifier	10.17605/OSF.IO/CQGTD
The Structuralization of Empiricism	DC1–DC5 divergence claims (§3)	10.17605/OSF.IO/J4GZ9
Update Integrity Standard (UIS)	Corruptor taxonomy and Empirical Ledger (§6–7)	10.17605/OSF.IO/DWM29
Records Across Nature, Life, and Mind	Persistence layer; record failure modes (§6)	10.17605/OSF.IO/7H6DY
T30 Prime Series, P0–P4	Hygiene-layer reporting standards (§4)	J2XSQ / Y678R / A6EJN / RN3TH / YHQ5F
Program Map	Layer architecture (§5)	10.17605/OSF.IO/CFASB
Two Worked Demonstrations	Support / failure / inconclusive reporting	10.17605/OSF.IO/Z7HY2

Kernel First, the Technical Notes, Methods papers, The Structuralization of Empiricism, the Update Integrity Standard, and Records are live OSF deposits. The T30 Prime series is now live, minted as a set, and the two companion documents — the Program Map and Two Worked Demonstrations — are live deposits; this Falsification Standard deposits alongside them as the third Architecture & Governance document.

UCT Library

This document is part of the Universal Collapse Theory library maintained by Jeremy C. Jones and HoldingLight LLC. It is the consolidated falsification standard companion to the Program Map. The per-claim conditions it gathers are stated in the Technical Notes, Methods papers, empirical demonstrations, and The Structuralization of Empiricism, all live deposits. Roadmap: universalcollapse.com/roadmap

AI Disclosure

AI tools were used to assist with manuscript preparation, drafting, organization, and editorial refinement. The underlying theory, structural decisions, analysis, and conclusions are the author's own.

Suggested Citation

Jones, J. C. (2026). Failure Modes and Falsification Standards for Universal Collapse Theory (v1.0). HoldingLight LLC. https://doi.org/10.17605/OSF.IO/TN7Z3