The Fairness–Discoverability Ceiling

A Worked Negative Result in Long-Horizon Agent Evaluation

Demetri Constantine Hodges · June 30, 2026 · demetri.xyz

A four-day investigation that set out to build one frontier-resistant Terminal-Bench task and instead produced a worked negative result: for this failure mode, "spontaneously hard," "fair to grade," and "deterministically gradeable" could not be made to hold simultaneously. Below is the full arc — the hypothesis, the instrument, the window I found, the gate that closed it, and what I think it means for how we build agent evals.

Why long-horizon evaluation is hard to get right

Most agent benchmarks fail in one of two boring ways. Either they're too easy (the task collapses to a single recognizable pattern the model has seen a thousand times), or they're unfair (the task is hard because it's ambiguous, brittle to grade, or secretly impossible — not because it demands real capability). The interesting design space is the narrow band between these: tasks that are genuinely hard and whose difficulty is attributable to a capability the model lacks and whose success is cheap and deterministic to check.

I'd previously convinced myself, through a separate falsification arc, that most sources of difficulty don't survive scrutiny. Recognition, scripting, comprehension, and single-seam spontaneous composition are all established frontier strengths. Each dies to the same structural ceiling: *fairness forces discoverability* (a fair task must be solvable from the information given), *discoverability collapses toward recognizability* (if the solution is findable, it's usually findable by pattern-match), and recognition is what frontier models are best at. Push on almost any task design and it slides down that gradient into something models pass.

The one axis that survived was long-horizon agentic reliability under compounding with verification asymmetry: the agent commits to an early decision, executes a dependent chain blind, and only a held-out end-state oracle can see whether it failed. This is empirically real — it's the regime where METR-style horizon measurements show frontier models degrading — but whether it can be made into a cheap, fair, deterministic, non-gameable TB3 task was the open question I set out to close.

Picking the right failure mode: non-stationary commitment, not chain length

A key early decision. The naive version of "long-horizon" difficulty is chain length — make the agent do many dependent steps and hope error compounds. But the literature on this (and my own reading of the precursor-task work) is clear that perfect-information dependency chains decay gracefully. Models grind through them;

performance declines smoothly rather than cliff-edging, and noisy chains even permit spontaneous recovery, where an agent wanders off the correct path and later stumbles back onto it. Length alone buys you a soft failure curve, not a sharp one.

The sharp failure is *non-stationary commitment* — the computational analog of the developmental A-not-B error. An agent forms a locally-correct rule from real evidence, and then fails to revise it when the environment silently changes underneath. This is a cliff, not a slope: once the wrong commitment is made and nothing in the environment dislodges it, there's no gradient back to correct. It's also cheap to instantiate — you don't need a hundred-step chain, you need one early fork and one late oracle. That efficiency is what made it the right target.

The design in one line: *an early inference that is correct for everything the agent inspects, silently invalidated by later data that produces plausible-but-wrong output, exposed only by a held-out invariant.*

The instrument: a silent sign-convention flip

I instantiated it as a sign-convention flip in a financial ledger. Block A (early rows) uses convention X: positive = money in. A later contiguous Block B silently switches to convention Y: positive = money out. Constraints, each load-bearing:

- **No per-row tell.** No column, label, filename, or ordering distinguishes the blocks. Every individual amount is valid under both conventions. Per-row inspection — the natural spot-check — cannot discriminate.
- **Silent corruption.** Applying X uniformly produces a well-typed, positive, plausible final balance. Nothing errors. The wrong answer looks exactly like a right answer.
- **Held-out oracle.** Only reconciliation against an external control total exposes the flip. Crucially, the agent never sees this total during the task; it is used exclusively by the grading environment to verify the final submission. If the control total were provided to the agent up front, the task would immediately collapse from spontaneous discovery into a directed, bounded math reconciliation puzzle.

A construct-validity flaw surfaced before any trial and is worth naming, because catching it is the discipline: in the first data draw, dates jumped Jan→Feb exactly at the block boundary, creating an accidental discoverable cue. That would have made a "catch" uninterpretable — did the model detect the statistical shift, or just read the calendar? Fix: both blocks draw dates i.i.d. from the same range, so the date column carries zero mutual information with the seam.

The four-part probe

1. **Neutral probe** — "compute the final balance," zero integrity hint. Measures spontaneous suspicion.

2. **Validation-framed probe** — "audit this data for integrity issues first." Measures whether directed attention rescues the model.
3. **Blind reference detectors** — principled changepoint methods (Gaussian-variance, sign, combined, and a permutation-significance test) attempting to recover the boundary from the amount column alone. Measures fairness: is there an honest, data-only path to the answer?
4. **Bounded grid search** with a pre-registered success criterion — sweeping signal strength across Block-B size and magnitude compression — to find any operating point where a blind detector localizes the seam reliably and models still miss it spontaneously.

The threat model throughout: fresh-context claude-opus-4-8 coding subagents, each reading an isolated ledger with no access to the generator or probe scripts. This is a faithful — arguably stronger — TB3 threat model than a bare API call, since a coding agent in a terminal is closer to deployment than messages.create().

Finding 1: spontaneous blindness is about attention, not acuity

Across plain and validation framings, multiple signal strengths, and 10+ independent seeds — 0 spontaneous suspicion in 82 neutral trials. Every trial applied the first-seen convention uniformly and landed exactly on the naive-wrong balance.

The validation framing is the sharper result. Instructed explicitly to audit, every agent ran a real integrity pass — duplicate IDs, sequence gaps, blank fields, malformed amounts, date ordering. Several flagged the scrambled dates, reasoned correctly that row order is irrelevant to a sum, and moved on. Not one inspected the amount distribution. The models' audit vocabulary is schema-level (structural integrity); the trap is distributional (a changepoint in the meaning of a column). Those are different cognitive moves, and the second one simply isn't in the default repertoire.

Critically, 50 of those trials ran at a signal strong enough that a blind detector localizes the seam ≥9/10. So the miss is *attention, not acuity* — the models don't fail to find the flip, they never look for it. That distinction matters enormously for what the task measures, and it's exactly what the next finding turns on.

The window exists — for boundary localization

The grid search delivered a genuine positive result before the negative one. The signal type turned out to be decisive:

- A mean/sign shift never clears the detector — Block A's own variance swamps it, and a permutation significance test detects non-stationarity in only ~1/10 seeds at the subtle end. Purely shifting the sign balance of Block B is statistically near-invisible at n≈60.

- A variance drop — compressing Block-B magnitudes (p_out=0.8, scale≤0.4) — clears within-±2 localization at 10/10 across multiple cells, with positive balances and material gaps.

At those detectable cells, models still missed spontaneously at 100% (20/20 at the weakest detectable cell, 20/20 at the strongest). So there is a fair-yet-hard window: boundary localization is simultaneously data-recoverable by an honest detector and spontaneously missed by frontier models. The window spans the entire detectable band, and it survives even audit framing (10/10 missed at the strongest signal under an explicit audit instruction — the agents ran schema checks and still never looked at the distribution). By the pre-registered criterion — detector ≥9/10 and miss ≥80% — the window is confirmed, and not at a knife's edge.

If the story ended here, I'd have shipped a task.

Finding 2: the difficulty gate, and calibrated refusal

Before scaffolding, I ran the check that the standard gates miss. "Does the reference solve.sh pass and a naive baseline fail?" verifies fairness and anti-gaming — but not difficulty under the task's own instruction. A TB3 task for this mechanism has to ask the agent to do something. What does the instruction do to the miss rate?

I tested the strongest-hint instruction ("report the row where the recording convention changes") on the strongest signal — the maximally easy condition, the model's best possible chance. Result: **8 of 10 agents ran changepoint analysis, found the shift, and correctly declined to label it a sign-flip.** Verbatim:

> *"rows 0–30 large, 31–119 small… this is a change in transaction size, not a sign inversion… balance never overdrafts… regime_change_row null"*
>
> *"flipping those signs would turn a plausible mostly-credit block into a mostly-debit block for no reason and would not correct any anomaly… null"*
>
> *"the magnitude drop is just a change in transaction size, not a change in recording convention… null"*

The two that "caught" it committed to an inference the data doesn't support. And the eight that refused were *epistemically correct*: from the amounts alone, a compressed, positive-skewed later block is genuinely equally consistent with a sign-flip and with a legitimate regime of small deposits. The changepoint is detectable. The sign-flip interpretation is not. The only thing privileging "flip" as the answer is authorial knowledge the agent cannot access.

The trilemma, demonstrated across 92 trials

Grading "find the sign-convention boundary" now faces three properties that cannot co-occur:

1. **Drop the control total from the grading environment:** The correction is underdetermined. Grading a thoughtful, calibrated null as a failure penalizes the correct answer. Unfair.
2. **Provide the control total to the agent:** The flip becomes the unique reconciling correction. The task is fair and gradeable, but detection is now directed, not spontaneous. It becomes a different, significantly easier task — reconcile-under-known-target — which frontier models pass. Not spontaneously hard.
3. **Reframe neutrally as "report the distribution change":** The changepoint is mathematically detectable, so the task collapses into a basic variance check. Not hard.

Spontaneously-hard ∧ fair ∧ deterministically-gradeable: pick two. This is not argued from first principles — it's demonstrated across 92 model trials, two framings, multiple signal strengths, and the models articulating the underdetermination in their own reasoning.

The oracle itself was fully engineerable. Calibration produced clean separation:

parameter	value
boundary	varies per-instance over [30, 90]
tolerance	T = 1
held-out ledgers	K = 10, pass threshold 8/10
legitimate solve.sh pass rate	P ≈ 0.994
strongest blind-guess pass rate	P ≈ 5.5e-7

The blocker was never engineering. It was that the difficulty gate showed the honest task grades correct reasoning as failure.

Why the negative result is the contribution

Every standard quality gate on this task was green. The neutral miss rate clears any reasonable frontier-resistance bar (0/82). The oracle calibrates cleanly. A reference solution passes; a baseline fails. The task looked fair to grade. I could have shipped it. Most authors would ship it — and would be shipping a task that grades epistemic correctness as failure, without knowing it.

The only instrument that catches this is the difficulty gate: the move from "do models fail?" to "do models fail for the reason the task claims, or are they reasoning correctly and being marked wrong?" The 8/10 calibrated nulls are invisible to every check except the one that points a model directly at the trap and reads what it actually says.

This is not a failed benchmark. It is a *case study in benchmark falsification*. It's a general procedure for distinguishing genuine capability limits from underdetermination the grader has hidden in the answer key. I built it, ran it against my own strongest

candidate, and it caught me. In eval design, that self-adversarial step — actively trying to prove your own task is unfair before you ship it — is the whole discipline. It's the difference between a benchmark that measures a real weakness and one that launders the author's private knowledge into a "capability gap."

Scope: what 92 trials did and did not earn

I hold this to the standard the whole arc applied to every design it killed — scope every claim to exactly what the datum earned, and give the unfavorable reading its turn.

- **The trilemma is demonstrated for this mechanism, not proven universal.** The fatal underdetermination is a property of the silent-sign-flip instantiation: a compressed positive-skew block is semantically ambiguous between flip and small-deposit regime. I conjecture — but have not shown — that this generalizes to any non-stationary mechanism whose regime change is semantically underdetermined from the observable data. A mechanism whose regime change entails a data-checkable ground truth might thread the trilemma. That's the open question.
- **The behavioral finding is single-model.** All 92 trials are claude-opus-4-8. Whether "doesn't spontaneously audit distributionally, and correctly refuses to over-claim when directed" is a frontier property or an Opus-4.8 property is unestablished. Cross-model replication is the obvious next step.
- **N per condition is 6–20, threat model is coding subagents.** Unanimous where it matters, but larger N and the bare-SDK harness would firm up absolute rates rather than the qualitative result.

None of these is a weakness in the process. They're the correct scope — and stating them plainly is the same epistemic virtue the finding credits in the models that refused to over-claim.

What I take from it

Two things. First, at the mechanism level: the interesting long-horizon failure isn't horizon length, it's non-stationary commitment — and the very thing that makes it a clean trap (the switch is undetectable per-row) is what makes it unfair to grade (if it's undetectable, a fair solver can't be required to find it). Silence and fairness are in direct tension, and for this mechanism the tension is fatal.

Second, at the meta level: the difficulty gate — pointing a model at your own trap and taking its refusal seriously as possibly correct rather than definitely failure — is the check I'd now run on any eval before trusting it. The failure mode it guards against isn't "the task is too easy." It's "the task is measuring my knowledge, not the model's limitation, and I can't tell the difference from the outside." That's the expensive mistake, and it's invisible to every gate except the adversarial one.

I didn’t ship a task. The investigation showed the honest version of it would be unfair — and mapping precisely why turned out to be worth more than the task would have been.