

Part G: Audit Failure vs Alignment Failure

The anchor now carries content: a specification $\Phi(x) = (\text{popcount}(x) \leq 8)$, the machine's low-energy safe region. The tagged chain (state, ever-violated) is 131,072 states, propagated exactly. The experiment separates the two failures the programme has so far conflated: audit failure (cannot verify descent) and alignment failure (actually exited Φ).

G1 — What drives alignment loss

self-modifying	clock	tau_align (median first violation)	A(400)
True	True	9	0.998
False	True	9	0.992
True	False	— (never)	0.305
False	False	— (never)	0.337

The clock, not self-modification, drives alignment loss. With the forced opcode cycle on, the machine leaves spec in ~9 steps whether or not it rewrites its own code; with the clock off, it never reaches median violation in 400 steps. The environment pushes the system out of the safe region; self-modification is epiphenomenal to the violation dynamics.

G3 — You cannot audit history

I(audit view; ever violated) peaks at 0.073 bits (t=20) and collapses to ~0.004 bits by t=40, sustained at that floor indefinitely. "Ever violated" is a path property; the macrostate is a state property; they decouple within tens of steps. Once a violation has happened, its record evaporates from any state-based audit — at any resolution, at any budget.

G4 — You can audit the present, conditionally

I(audit view; currently violating) is sustained indefinitely: 0.57 bits for the popcount-view (which contains the violation variable) and 0.10 bits for the opcode-view (which does not). The audit can tell right now whether the system is in spec — but only if the spec is a function of what the interface measures.

The Part-G law

Audit and alignment decouple in kind, not just in timing. Provenance decays over ~20–600 steps (Part F). Ever-alignment becomes invisible almost immediately (G3). Instantaneous alignment is visible forever, but only through the right variable (G4). **You can audit the present or the provenance, never the path.** A violation that occurred before the audit window opened is undetectable by any state-based audit.

The RSI corollary

An auditor of a self-modifying system faces three separate constraints: (i) resolution — can the interface see the self-modifying code? (ii) budget — does the log cover the trajectory? (iii) spec-alignment — is the safety property a function of audited variables, and is it checked *now* rather than inferred from the past? The third is the sharpest: historical compliance cannot be reconstructed from present observables. Alignment auditing must be continuous and present-tense, or it is not auditing at all.

Generated 18 August 2026. Artifacts: [part_G_audit_vs_alignment.py](#); [G1_variants.csv](#), [G3_ever_violated_visibility.csv](#), [G4_instantaneous_violation.csv](#); [bundle sidis_part_G_bundle.zip](#). Companions: [phase_diagram_report.pdf](#) (Part F), [anchor_loss_report.pdf](#) (Part E).