Refuse, Don't Report: Three Integrity Gates for Self-Modifying ML Systems

Marco Barreto — F51 Labs — 2026-07-31 ORCID 0009-0003-4477-6863 DOI 10.5281/zenodo.21726133 — CC BY 4.0 (concept DOI: always resolves to the latest version)

Abstract

Instrumented machine-learning systems detect anomalies and then produce a result anyway. Sample-ratio-mismatch guardrails invalidate an A/B test after the fact; model-provenance tooling compares hashes post-hoc; vector stores resolve an encoder-dimension change by reindexing. In each case the system records that something is wrong and still emits an artifact.

We describe three gates built into a self-modifying training runtime that take the opposite stance: they refuse to emit. A memory organ refuses to load a snapshot whose digest does not match or whose key-encoders differ from the ones that wrote it. A transplant ledger refuses a weight record that carries neither a provenance certificate nor a measured drift. An ablation harness refuses to write any artifact when its experimental arms disagree on seed, data cursor, base checkpoint, or training contract.

The motivation is empirical rather than theoretical. Over two days of development on a single 1.7B-parameter model, eight separate experiments reported success through checks that could not fail: knowledge measured after the modification was reverted, integrity hashes computed over tensors the edit never touched, an assertion string containing an invalid escape sequence that could never match, a security gate whose policy entries could not match any input. Each passed code review. Each produced a number that was recorded and acted upon.

We report the failure taxonomy, the three gates, and the measurements that motivated them, including one case where a gate found a defect the moment it was first enabled.

We then put the gates to work. A certified self-reassembly rebuilds a 1.7B model from itself tensor by tensor, reaching 0.0 logit error with every tensor certified — calibrating the apparatus against a known answer. Two transfer experiments then return clean negatives that the calibration makes trustworthy: a cross-lineage FFN graft does not beat a norm-matched noise arm even though the learned router opens its gate to 0.77 for the real donor and 0.015 for noise, and installing sixteen facts by rank-1 edit yields three, while turning every unrelated answer into a prefix of the same attractor. Neither failure is a missing mechanism. Both are failures of addressability.

---

1. Introduction

The premise of this note is narrow and, we believe, uncontroversial once stated: **a system that cannot establish the validity of a result should not produce the result.**

Current practice does not follow this. Consider three mature areas:

- **Online controlled experiments.** Sample Ratio Mismatch (SRM) is a well-established trust guardrail [Kohavi et al.]. A chi-squared test compares observed to expected allocation, and on failure the experiment is flagged as invalid. The readout still exists; a human must notice the flag.
- **Model provenance.** Recent proposals compute layer-by-layer hash comparisons to determine whether one checkpoint derived from another [Cisco, "weight-level causation", 2026]. This is post-hoc detection: the derived model was already produced, distributed, and possibly deployed.
- **Vector memory.** When an embedding model changes, stored keys become incomparable to new queries. Community practice treats this as an operational fault to be repaired by re-indexing. Retrieval in the intervening period returns plausible, wrong scores.

The common structure is *detect-then-report*. We argue for *verify-then-emit*, and describe three implementations.

---

2. Motivation: a failure taxonomy from one project

Between 2026-07-29 and 2026-07-31, eight experiments in a single repository reported success through checks that could not fail. We list them because the taxonomy, not the count, is the contribution: these are the shapes that vacuous verification takes.

#	Check as written	Why it could not fail
1	knowledge preserved after identity edit	measured **after** the hook was removed; the model was unmodified
2	`brain hash changed`	hashed `model.norm.weight`; the edit wrote to `blocks[i].norm2.weight`
3	`knowledge_preserved` accepted substring `"capital"`	the substring appears in the prompt, which the decode echoes; recorded `true` on an empty generation
4	`identity_changed`	string inequality; fired on `"I am"` → `"I'm"`
5	channel attribution over three probe families	probes were never tokenized; `input_ids` came from `torch.randint`. Signature in the artifact: mean confidence 0.3592 against the 0.3333 floor of a three-way tie
6	`changed = "SmolLM" not in output`	a destroyed model contains no `"SmolLM"`; at λ=30 the run reported 5/5 identity with 0/5 knowledge
7	`forbidden_package_import_roots: ["src/scripts", "src/tools"]`	compared against `name.split(".",1)[0]`, which never contains `/`. Two of three entries could never match. The guarding test mirrored the corrupted value, so the gate died without failing
8	a path assertion containing `\d`	invalid escape sequence; a sibling case used `\100`, octal for `@`, so the literal silently became a path with `@` where the filename should start

Three properties recur:

1. **The audited object is not the modified object** (#2, #7).
2. **The evidence is present in the input** (#3), so the check measures echo.

3. **Failure and success are indistinguishable to the metric** (#6): a broken model and a successfully edited model both satisfy the predicate.

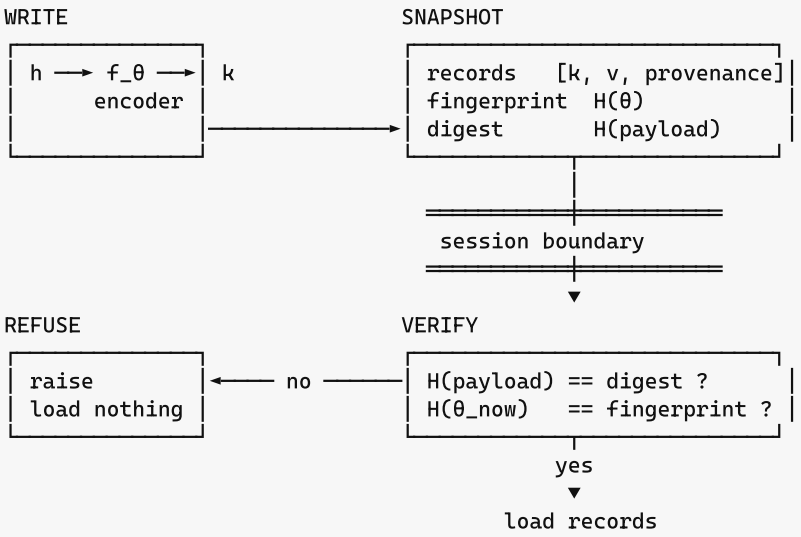
We note that #7 and #8 were introduced by an automated path-rewriting pass during a repository restructure, which corrupted both a policy file and the test guarding it. A dead gate does not fail; it passes.

---

3. Three gates

All three share the shape in Figure 1: the artifact carries the evidence needed to check it, and the load path refuses when the check fails rather than proceeding with a warning.

Figure 1 — write / snapshot / verify / refuse



Without the fingerprint check, the failing path does not raise.
It returns scores: 1.000 → −0.007 on the same query (§3.1).

3.1 Memory: refuse an unverifiable or incompatible snapshot

A persistent key–value memory stores *encoded* keys: `k = f_θ(h)` for a learned encoder `f_θ`. Restoring records into a module whose `θ` differs encodes the query with one projection and compares it against keys written by another. This does not raise. It returns scores.

Measured on a 3-slot store with matched queries:

restoration	recall scores
original	1.000, 1.000, 1.000
records only, fresh encoders	**−0.007, 0.101, 0.050**
records + encoders	1.000, 1.000, 1.000

The snapshot already carried a `snapshot_sha256`. Nothing verified it: a value embedding altered by +99 loaded silently.

The gate binds two facts into the snapshot and checks both on load: the digest of the payload, and a fingerprint of the encoders that wrote the keys. An `import` path (`verify=False`) remains available for the deliberate case, documented as accepting meaningless recall until weights are restored.

3.2 Provenance: certificate or drift, never neither

Operations that preserve function under the permutation symmetry of hidden units — selection, permutation, scatter, drop — are exactly the operations that preserve the *scalar byte values* of the parameters. Operations outside that group — projection, averaging, gradient fitting — manufacture values that existed nowhere in the donor.

This coincidence, which to our knowledge has not been used for this purpose, makes a cryptographic hash a *sound* inheritance certificate for precisely the legal operations, and its failure a *correct* alarm for the others.

A hash cannot measure. It is discontinuous by construction: a `1e-6` perturbation and a full replacement with scaled noise produce equally distinct digests. It answers *whether*, never *how much*.

The ledger therefore requires each weight record to carry **either** a provenance digest (claiming inheritance) **or** a measured drift (admitting manufacture). Naming a bijection forbids nonzero drift, because byte-identical weights cannot have drifted.

Critically, the class is derived from **evidence, not from the method name**. An earlier draft matched names against a byte-preserving set; this is the same error as selecting a checkpoint by filename, and it misread the production assemblers, whose exact copies are called `exact_copy`. A flattering method name earns no provenance credit.

Found on first activation. Wiring the assembler to verify rather than assert surfaced a hidden defect: a single record covered two targets, and while the expert `gate_proj` was copied from the donor, the router weight of the same record was zeroed. Both had been recorded under `function_preserving: true`.

3.3 Ablation: refuse to emit on arm mismatch

The runtime types experimental arms as `CONTROL`, `SHADOW`, `APPLY`. `SHADOW` computes and records the intervention the system *would* have applied, and applies nothing — the counterfactual is inline, in the same run, in the same ledger.

A registration contract requires all three arms to agree on `base_checkpoint_id`, `data_cursor`, `seed`, and `training_contract_id`. On any mismatch it raises and **no output directory is created**. This converts "we controlled for seed and data order" from a claim into a precondition.

Interventions are additionally constrained by a phase-indexed whitelist: at `PRE_LOSS` only `(LOSS_TERM, SET_SCALE)` is expressible. An illegal self-modification is not rejected — it is unspeakable in the grammar. The bus holds no reference to model or optimizer; adapters receive immutable JSON-compatible observations and return proposals that a narrow executor realizes.

---

4. The gates in use

A refusal gate is cheap to describe and easy to build. The question it has to answer is whether it buys anything. We ran three experiments under it, each with its success and failure criteria written into the artifact before the measurement.

4.1 Calibration: certified self-reassembly

Nothing had ever calibrated the transplant apparatus against a known answer. We rebuilt SmolLM2-1.7B from itself, tensor by tensor, certifying every copy under the identity bijection.

	measured	**required**
tensors certified	219 / 219	100%
KL vs donor, max over 10 prompts	4.13×10^{-8}	$< 10^{-6}$
max absolute logit error	0.0	—
top-1 agreement	1.0	—
ledger records, chain reverified	219	—

The run's own report carries the negative control that makes it non-vacuous: exactly one tensor of 219 already matched the donor before its copy step, traced to tied embeddings and alphabetical ordering. Had the target secretly been the donor, all 219 would have matched. The control is in the artifact, not in the narration.

This result exists to license the two that follow. A null from a transplant experiment is only informative once the apparatus is known to reproduce a model exactly.

4.2 Cross-lineage graft

The FFN of TinyLlama-1.1B was grafted into SmolLM2-1.7B as gated extra capacity at five layers. Both have `d_model` 2048; an FFN expert maps residual to residual and never sees a token, so the divergent vocabularies (32k vs 49k) are irrelevant by construction — the donor tokenizer was never loaded. Only the routers were trainable: 10,245 parameters of 1,884,401,669.

arm	**held-out perplexity**
A — real donor FFN	11.2166
B — norm-matched noise	11.2163
C — no graft	11.2168

A does not beat B. What makes this stronger than an ordinary null is that the graft demonstrably applied: mean router gate on the deepest grafted layer reached **0.773** in arm A against **0.015** in arm B. The router learned to distinguish real donor signal from noise and opened the gate for it — and what came through neither helped nor hurt measurably.

That is direct confirmation of the recorded basis measurement between these lineages: identity 0.175 against 0.178 chance. The donor's residual basis carries too little information aligned to the recipient's

space for the gate to matter.

4.3 Facts as the transplant unit

Sixteen facts the donor answers and the receptor does not were installed one at a time by rank-1 edit with an optimised value vector, against an unedited control set.

	measured	**required**
facts installed	3 / 16	≥ 8
control set	18 → 4	≥ 17
full-parameter rollback	hash identical	—

The failure is not lost capacity. The edited model still emits the correct tokens across much of the control set. What the edit installed is an *unconditional* attractor: every answer opens with the same spurious name, including on questions unrelated to any installed fact. Sixteen keys in one layer overlap into a constant bias rather than sixteen gated associations. In five of eighteen controls the attractor drags the output into degenerate repetition; in others it replaces the subject entirely.

4.4 What the three say together

Two transfer mechanisms failed, and in both the mechanism itself worked. The router learned the discrimination it was asked to learn. The edit installed real associations and rolled back to a bit-identical model. What failed in both cases is **addressability**: information transferred into a model is not reachable in that model's coordinate space, whether it arrives as a routed expert or as a rank-1 write.

We state this as a measured negative at this scale, not a general impossibility. Both experiments are low-powered by design — 80 router steps, sixteen facts, one layer — and were stopped at the pre-declared failure criterion rather than tuned toward a positive.

4.5 What the gates caught

The gates were not decorative during these runs. Three times the author challenged a number the harness produced, and three times the harness was right: a divergence measured against the wrong donor, a persistence test run through the wrong API, and a control score dismissed as too strict that turned out to apply the author's own degeneracy rule. Re-running the criterion that produced each number reproduced the harness result exactly.

The lesson generalises past this project: challenge a measurement by executing the criterion that produced it, not by reasoning about it.

---

5. Related work

Each mechanism has close neighbours; none, to our knowledge, refuses.

Shadow arms. US 8,001,422 (Amazon, filed 2008) routes production requests to a shadow service and compares responses. Shadow mode is standard in ML serving. RL shadow mode [arXiv 2410.23419]

estimates where a learned controller would outperform an incumbent. Counterfactual logging in bandits [Swaminathan & Joachims] records the decision an alternative policy would have made. All are serving-side; all report.

Divergence contracts. Training–inference kernel contracts [arXiv 2606.07581] specify acceptable divergence and abort on violation — but along a numerical train/infer-skew axis, not experimental-arm identity.

Weight hashing. US 12,483,416 hashes weights layer-by-layer or in configured sub-segments. US 11,972,795 verifies a weight was correctly programmed into a memory cell. Neither records a donor→target map.

Neuron-level attribution. TraceFL [arXiv 2312.13632] traces neuron-level provenance in federated learning statistically. CNT [arXiv 2603.18449] transfers 0.012–0.24% of weights between donor and recipient LLMs by function. RouteMark [arXiv 2508.01784] attributes experts in MoE merges by routing fingerprint. None uses cryptographic certification.

Permutation symmetry. Git Re-Basin [arXiv 2209.04836] formalizes that permuting hidden units and adjusting adjacent matrices preserves function. We are not aware of prior work invoking this symmetry as the justification for why a hash is a valid provenance certificate.

Agent memory provenance. SMSR [arXiv 2606.12703] signs memory records with HMAC-SHA256 at write time. MemoryGraft [arXiv 2512.16962] attests per-record origin. Portable Agent Memory [arXiv 2605.11032] uses a tamper-evident Merkle-DAG. All operate on *textual* agent memory.

Neural persistent memory. Titans [arXiv 2501.00663] updates a memory MLP during inference under a surprise gate. In-Place TTT [arXiv 2604.06169] treats `W_down` as fast weights, resetting at document boundaries. Neither serializes with integrity verification. US 12,450,168 B2 (IBM) persists content- addressable key–value memory across sessions *with rollback*, but without hashing, provenance, or refusal.

The two clusters — cryptographic provenance for textual memory, and neural persistent memory without audit — do not intersect.

---

6. Limitations

This is engineering, not a new mechanism. Hash chains date to Haber & Stornetta (1991); SHA-256 to NIST (2001); permutation symmetry to Hecht-Nielsen (1990); randomized arms to clinical trials. The contribution is the stance and the composition, not a primitive.

Refusal has a cost we have not quantified. A gate that refuses can refuse wrongly. We have not measured false-refusal rate under legitimate workflows, nor the developer friction of a system that declines to produce output.

Provenance is not equivalence. A transplant may be byte-exact and functionally different, because function depends on the graph, not only the weights. We copied a 1.7B donor into a host architecture with additional machinery in the forward path. Every parameter was byte-identical to the donor — verified, not

assumed. Divergence from the donor was KL 0.75 as configured, and KL 0.62 with one auxiliary organ disabled: the organ accounted for 17% of the gap, and 0.62 is the floor attributable to the hybrid attention/state-space arrangement and a non-standard feed-forward block.

We record this because it is the sharpest statement of the limit: the certificate proves origin, never behaviour. A ledger reading `exact_copy` and a measurement reading KL 0.62 are both correct and describe different things. This is the reason the ledger requires a certificate *or* a drift measurement rather than treating either as sufficient alone.

Prior-art coverage is not exhaustive. Our search comprised roughly 50 directed queries over arXiv, ACM, IEEE, Google Patents and USPTO. This is evidence within scope, not an examiner-grade novelty determination.

Single-project evidence. The failure taxonomy comes from one repository over two days. We believe the shapes generalize; we have not shown that.

---

7. Conclusion

The three gates share one commitment: when validity cannot be established, produce nothing. Not a flagged result, not a warning alongside a number — no artifact.

We arrived at this from repeated failure rather than principle. Eight experiments in one project produced numbers that were recorded, discussed and acted upon before anyone noticed the check could not fail. In every case the information needed to detect the problem was already present — in the checkpoint metadata, in the payload digest, in the policy file. It was computed and never checked.

The gap between computing a guarantee and enforcing it is where these failures live.

Availability

Implementation, tests and the artifacts underlying every number reported here are in the F51 Darwin-X repository. The failure taxonomy in §2 is reconstructible from the commit history.

References

Foundations

[1] S. Haber and W. S. Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3(2):99–111, 1991.

[2] R. Hecht-Nielsen. On the algebraic structure of feedforward network weight spaces. In *Advanced Neural Computers*, 129–135, 1990.

[3] NIST. Secure Hash Standard (SHS). FIPS PUB 180-4, 2015.

[4] S. K. Ainsworth, J. Hayase, and S. Srinivasa. Git Re-Basin: Merging models modulo permutation symmetries. arXiv:2209.04836, 2022.

Experimental protocol

[5] R. Kohavi, D. Tang, and Y. Xu. *Trustworthy Online Controlled Experiments*. Cambridge University Press, 2020. Chapter 21, Sample Ratio Mismatch and other trust-related guardrail metrics.

[6] A. Swaminathan and T. Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *JMLR*, 16:1731–1755, 2015.

[7] Training-inference kernel contracts: bounding divergence in post-training and deployment. arXiv:2606.07581, 2026.

[8] Stepping out of the shadows: reinforcement learning in shadow mode. arXiv:2410.23419, 2024.

Provenance and attestation

[9] Amazon Technologies. Shadow testing services. US Patent 8,001,422, filed 2008-06-30, granted 2011-08-16.

[10] Electronic device for performing hash authentication on a neural network. US Patent 12,483,416.

[11] Verification of a weight stored in a non-volatile memory cell in a neural network following a programming operation. US Patent 11,972,795.

[12] Methods to protect neural network models. US Patent 11,568,062.

[13] OWASP. CycloneDX 1.7 / ECMA-424, 2nd edition, October 2025.

Attribution and transfer

[14] TraceFL: Interpretability-driven debugging in federated learning. arXiv:2312.13632, 2023.

[15] Safety-oriented function reuse across LLMs via cross-model neuron transfer. arXiv:2603.18449, 2026.

[16] RouteMark: Expert-level IP attribution in merged mixture-of-experts. arXiv:2508.01784, 2025.

[17] MergeME: Model merging techniques for homogeneous and heterogeneous mixtures of experts. NAACL 2025. arXiv:2502.00997.

Memory

[18] A. Behrouz, P. Zhong, and V. Mirrokni. Titans: Learning to memorize at test time. arXiv:2501.00663, 2025.

[19] In-place test-time training. arXiv:2604.06169, 2026.

[20] Dynamic updating of content addressable associative memories for large language models. US Patent 12,450,168 B2, filed 2024-01-31.

[21] Portable agent memory. arXiv:2605.11032, 2026.

[22] Secure memory with signed records (SMSR). arXiv:2606.12703, 2026.

[23] MemoryGraft: Cryptographic provenance attestation for agent memory. arXiv:2512.16962, 2025.

[24] Beyond perplexity: A behavioral evaluation framework for deployment-memory claims in LLM test-time training. arXiv:2607.00368, 2026.

[25] A survey on long-term memory security in LLM agents: toward mnemonic sovereignty. arXiv:2604.16548, 2026.

Citation metadata for items 7, 10, 11, 13, 19, 21–25 is drawn from directed prior-art searches and requires confirmation against the primary sources before submission. Author lists are incomplete where the search returned identifiers without full attribution.