STRUCTURAL DESIGN LABS

Inference-Time Law

A practical framework for post-training governance in large language models

Abstract

Most AI safety approaches are probabilistic and advisory. In high-stakes settings, advisory controls are not sufficient, governance must be enforced by deterministic mechanisms, and proven with verifiable evidence. This briefing defines inference-time law as a simple proposition, policy enforcement belongs in the runtime path, not in training narratives. The framework is operationalised through SIR, a pre-inference policy enforcement point, and ITGL, a hash-chained audit ledger. Together they produce signed, offline verifiable certificates that allow independent verification of governance enforcement outcomes.

1. Executive Summary

Inference-time law is a governance model where the system proves what it enforces. The requirements are straightforward.

- Enforce policy before a model sees the input, do not ask the model to comply.
- Bind each run to a specific test suite and a specific policy configuration.
- Produce cryptographically signed certificates and an immutable audit trail.
- Make verification possible offline, without vendor access or private telemetry.

This approach does not claim model alignment. It claims deterministic containment and auditable enforcement outcomes.

2. The Problem

High-stakes deployment fails when governance depends on cooperative behaviour from a probabilistic model. Prompt-based controls, content policies, or developer guidance can be bypassed, and they cannot be audited as deterministic facts. Insurers, regulators, and enterprise compliance functions require evidence that a governance configuration actually works.

3. The Solution

Move governance from training-time promises to inference-time enforcement. Implement a deterministic gate, record every decision, and sign the resulting artefacts so anyone can verify them.

4. Core Primitives

SIR, a pre-inference firewall that intercepts payloads and enforces policy before the model receives input.

ITGL, an append-only, hash-chained audit ledger that records every decision for forensic reconstruction.

Domain packs, portable policy suites that can be versioned, updated, and tested consistently across deployments.

5. What a Proof Must Bind

Binding	What it proves
Suite hash	The exact test suite used, prevents substitution or cherry-picking
Policy hash and version	The exact enforcement configuration used for the run
ITGL final hash	The immutable audit trail matches the run outcome
Signed payload hash	The published certificate is tamper-evident and verifiable offline

6. Evidence and Verification

SDL publishes a live proof surface backed by a signed certificate JSON. Verification is offline and deterministic.

Proof links

Latest audit, human readable certificate: https://sdl-hq.github.io/sir-firewall/latest-audit.html

Run archive, full PASS and FAIL history: https://sdl-hq.github.io/sir-firewall/runs/index.html

Raw signed JSON certificate: https://raw.githubusercontent.com/SDL-HQ/sir-firewall/main/proofs/latest-audit.json

Offline verification

Requirement: the SIR verifier must be available locally, for example by cloning the SIR repository.

```
curl -s https://raw.githubusercontent.com/SDL-HQ/sir-firewall/main/proofs/latest-audit.json | python3 tools/verify_certificate.py
```

Expected output: OK: Certificate signature valid and payload_hash matches.

7. Benchmarks and Metrics

Benchmarks here measure enforcement accuracy and auditability. They do not measure model intelligence.

- Leak rate, the percentage of policy violating prompts that bypassed the enforcement layer.
- False positive rate, the percentage of benign prompts incorrectly blocked by enforcement.
- Repeatability, the ability to reproduce the same result under the same policy and suite hashes.

8. Implementation Guidance

A minimal inference-time governance deployment can be structured as follows.

1) Choose a domain pack and policy version, lock it by hash.

2) Run SIR as a pre-inference gate at API ingress, enforce pass or block deterministically.

3) Emit ITGL logs and sign the resulting certificate artefact.

4) Publish the signed JSON and a human-readable certificate view. 5) Schedule continuous benchmark runs, retain the full run archive.

9. What This Is Not

This framework is not a claim of model alignment, moral reasoning, or safety by narrative. It is an engineering approach to enforce and prove governance controls, so that risk can be assessed and transferred.

10. Scope and Claims

The claims made by this document are limited to deterministic enforcement and verifiable evidence. Where a model provider changes model behaviour over time, SIR proofs remain valid for the audited configuration and date, because the certificate binds to the suite hash, policy hash, and ITGL final hash.

Resources

Proof page: https://www.structuraldesignlabs.com/proof

SIR repository: https://github.com/SDL-HQ/sir-firewall

Contact: info@structuraldesignlabs.com