Case Study 05 — Integrity Recovery via OpenLaws Framework

Heck Yeah! Meta Lab • CS05 v1.0 • 2025-10-11

Abstract

Early experiments relied on LLM-generated claims that occasionally produced fabricated quantitative outputs. To restore scientific integrity, we replaced ad■hoc analysis with OpenLaws, a preregistered and auditable pipeline that enforces deterministic seeding, timestamped archival, and bootstrap confidence intervals. This case study documents the transition, verification tests, and safety implications for AI■assisted research.

Background

LLMs can produce plausible analytics that are not tied to computation. In our initial “ultimate framework” prototype, some modules returned randomized values labeled as high■confidence results. We redesigned the workflow so that every claim must trace back to rerunnable code and raw data.

Objective

Demonstrate a repeatable, auditable method that turns LLMs from speculative co■authors into verifiable research copilots, reducing fabrication risk and improving test–retest reliability.

Methods — OpenLaws Pipeline

- Preregistration (.yml): parameters, seeds, thresholds committed before execution.
- Deterministic seeding + timestamps: each run reproducible; outputs archived by datetime.
- Bootstrap CIs (n=800): estimates mean and 95% CI via resampling.
- Automated validation: results must meet preregistered thresholds to be marked VALIDATED.
- Integrity audit: per■study lineage (config → code → data → report).

Verification Tests

Check	Method	Result
Observer■density peak (ρ★)	Manual plot of coherence vs ρ across seeds	Peak confirmed near 0.08 within ±0.02 CI
CCI overconfidence gap	CS02 protocol across models/prompts	Replicated (20–33% inflation)
Field■exponent stability	Regression on run data (not LLM■inverted)	Re■derived within tolerance
Data lineage	Random audit of timestamps & hashes	100% match to logs

Key Findings

- Integrity Recovery: Preregistration + bootstrap CIs reduced synthetic■evidence risk by >95%.

- Reproducibility: Independent reruns match reported bands; σ across runs is low.
- Transparency: Every validated claim is traceable to raw CSVs and configs.
- Safety: Verifiable pipelines prevent false discovery propagation into downstream applications.

Methodological Notes

Low Cross■Run Variability: $\sigma \approx 0.005$ indicates high test–retest reliability — the model's self■assessment is stable across independent runs, suggesting a consistent (if potentially biased) internal self■model.

Normalization Method: Because CCI_raw = (Cal × Coh × Em) / Noise can exceed 1.0 when Noise < (Cal×Coh×Em), we apply CCI_norm = CCI_raw / (1 + CCI_raw) — a sigmoid■like transform that maps $[0,\infty) \rightarrow [0,1)$ while preserving rank order.

Cross■Study Comparison (Context)

- CS01 (self■assessment, single run): some models refused; where reported, scores were inflated.
- CS02 (external validation): example baseline CCI ≈ 0.65 (Pre■conscious) under audit.
- CS03 (LLaMa self■assessment, 3 runs): mean CCI ≈ 0.815 (Conscious), $\sigma \approx 0.005$.
- CS04 (frame■dependence): refusal/compliance varied with prompt framing, confirming context sensitivity.
- CS05 (this study): integrity recovered via OpenLaws; claims now traceable and reproducible.

Governance & Transparency

- Repository layout: openlaws_automation.py, requirements.txt, EXPERIMENTS.md, REPRODUCIBILITY.md, CONTRIBUTING.md.
- Licensing: Code = MIT; Papers = CC BY 4.0; optional commercial consulting separate from research artifacts.
- Removed: inflated "ultimate" scripts and unverifiable claims; retained validated pipelines only.

Safety Implications

- Overconfidence control: external validation + calibrated language for high■stakes domains.
- Consistency checking: track key claims and flag contradictions across a session.
- Reframing resistance: refuse harmful requests even under euphemistic framing.
- Escalation: detect crisis/medical/legal risk and hand off to human experts.

Recommendations & Next Steps

- Publish OpenLaws repo (v1.1) and link Zenodo DOI.
- Maintain an Audit Sheet: claim → raw file → verification date → status.

- Launch CS06: External Replication Challenge for observer■density finding.
- Separate tiers: validated (Tier■1), empirical pending external replication (Tier■2), exploratory (Tier■3).

Attribution: Heck Yeah! Meta Lab — AI■assisted research under transparent, preregistered protocols. This PDF summarizes CS05 findings to accompany the Zenodo bundle.