Sheaf-Guarded Updates: Streaming Structural Verification for Evolving Agent State

Anonymous Authors[1]

Abstract

Self-evolving scientific agents need update-time verification: does the agent's internal state remain coherent after a local self-modification? We model agent state as a growing graph equipped with a cellular sheaf and use coboundary checks, H^1 obstruction, Dirichlet energy, and Laplacian spectra to detect represented structural contradictions before commit. Under explicit policy and regularity assumptions, we prove conditional Lyapunov stability results for density and spectral-gap dynamics. A cellular decomposition localizes cohomological recomputation to $O(1)$ amortized per-edit cost under bounded-cell and cached-assembly assumptions, enabling maintenance at $V{=}5$M vertices with 35 μs median per-edit latency and zero assembled-cohomology drift on a single commodity machine. On ProofDAG, a controlled proof-dependency coherence test, per-edge coboundary testing achieves exact separation (F1 $=$ 1.000) under exact representation and F1 $\geq$ 0.94 under 14% relative noise. A 990-lemma study with live LLM proposals across 10 mathematical domains shows that coarse natural-language loading causes structural aliasing: valid and contradictory coboundary distributions overlap, identifying loader fidelity as the binding constraint for autonomous commit decisions. A Lean/mathlib benchmark confirms the exact-loader path on 507 declarations and 954 kernel-certified dependency edges across 6 namespace groups, with zero clean residuals (max 2.61×10^{-16}) and all 62 controlled corruptions detected. We present an implemented demonstration of bounded-cell sheaf-cohomological verification under streaming graph mutations at multi-million-vertex scale.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Submitted to the AI4Math Workshop at ICML 2026.

1. Introduction

Self-evolving scientific agents combine mathematical reasoning, tool use, proof search, hypothesis generation, and structured state updates.[1] As these systems accumulate claims, evidence links, proof obligations, and intermediate results, the central question becomes whether the evolving state remains coherent after each local update. Neural generators do not by themselves certify that an evolving state remains compatible with its encoded constraints; generation is no longer the sole bottleneck, state maintenance is. Each tool result, memory write, dependency insertion, or proposed lemma is a state transition that must be verified before commit. Large populations of persistent agents require verification whose cost is tied to update locality rather than total accumulated state. This paper addresses that problem with a deterministic topological substrate that detects and localizes coherence failures as they occur, targeting two themes in the AI4Math call: *verification and measurement* for self-evolving scientific agents, and *scientific problem solving* at the level of theoretical computer science.

We model the agent state as a growing graph equipped with a cellular sheaf (§3), where a closed-loop governor uses obstruction signals to trigger certified updates, closing the "neural proposes, sheaf verifies" loop. All restriction maps satisfy a bounded-gain condition enforced by deterministic rescaling (§3.2), placing the system in the near-identity regime of First & Kaufman (2024).

Our contribution is a bounded-gain cellular-sheaf verification substrate that checks whether proposed updates preserve coherence. We separate claims into three categories: architectural invariants enforced by construction (bounded restriction-map gain, bounded-cell recomputation), conditional theorems establishing stability under stated update-policy assumptions, and empirical measurements of scale, drift, noise tolerance, and loader fidelity. Specifically, we provide: (1) two conditional Lyapunov stability theorems for the edge-vertex ratio and Laplacian spectral gap, proved via Borkar stochastic approximation, with a stability corol-

[1]Portions of this manuscript were refined with AI assistance for clarity, LaTeX formatting, and workshop framing. The authors take full responsibility for all content and mathematical correctness.

lary for certified near-identity sheaves as a restricted positive result compatible with the absence of a universal Cheeger inequality for arbitrary sheaves; (2) a density-aware governor and Purity Gate that supply key stability assumptions by construction; and (3) a cellular decomposition that localizes verification to bounded cells, reducing per-edit cost to $O(1)$ amortized in graph size, validated at $V{=}5$M with 35 μs median per-edit latency. All results are from a fully implemented system. The resulting picture is deliberately two-sided: under exact representation, the substrate gives sharp update-time verification; under lossy natural-language loading, the same machinery exposes where mathematical structure has been erased. For AI4Math, the contribution is not a theorem prover or tactic generator, but a pre-commit layer deciding when to commit, block, or decompose.

Scope. The substrate verifies *constraint compatibility*, not *semantic truth*; detecting logical falsity requires a loader that encodes domain semantics. We quantify loader fidelity under noise (§5.8) and live LLM proposals (§5.9).

2. Related Work

Sheaf ML. Hansen & Gebhart (2020) introduced sheaf neural networks; subsequent work developed Cheeger-type inequalities (Bodnar et al., 2022), connection Laplacian variants (Barbero et al., 2022), bundle NNs (Bamberger et al., 2025), and sheaf-theoretic attention (Hu et al., 2026), all on fixed graphs using H^0. Ayzenberg et al. (2025) survey the field and note that H^1 and higher cohomology remain unexplored as operational signals. Hernandez Caralt (2026) shows identity restriction maps match learned Laplacians on standard heterophilic benchmarks, suggesting non-trivial sheaf structure may be unnecessary for H^0-based tasks. Our system uses H^1 obstruction as the primary feedback signal on growing graphs.

Spectral sheaf theory. Hansen (2019) developed spectral theory for cellular sheaves. First & Kaufman (2024) proved no universal Cheeger inequality extends from graphs to arbitrary sheaves but established positive results for restricted classes. Our near-identity regime ($\sigma_{\max} < 1$) falls between their positive results for constant sheaves and negative results for locally constant sheaves. Bandeira et al. (2013) proved a Cheeger inequality for the graph connection Laplacian. Wei & Wei (2025) developed persistent sheaf Laplacians with spectral bounds across filtrations; Le Vu Anh et al. (2025) established Lipschitz bounds for persistent Laplacian eigenvalues, complementing our Lemma 1.

Growing graphs and Lyapunov methods. Barabási & Albert (1999) studied preferential attachment. Gross & Blasius (2008) reviewed adaptive coevolutionary networks. Existing Lyapunov results in learning systems (Revay et al., 2024; Richards et al., 2018) operate on fixed topologies. Borkar (2008) developed the stochastic approximation framework we use; Borkar (2026) extended it to non-Markovian environments. Our Lyapunov analysis adapts this framework to a growing graph-sheaf pair, where the topology itself evolves.

Formal verification for AI. Dalrymple et al. (2024) proposed a framework for guaranteed safe AI requiring formal verifiers. In AI4Math, verification typically attaches to final artifacts: proof scripts, formalized statements, or solutions. We target a different object: the mutable internal state of a scientific agent before final output.

Self-evolving scientific agents. Recent AI4Math systems combine language models, proof assistants, tool use, and iterative search. Most work focuses on generation: theorem proving, tactic synthesis, autoformalization, or mathematical reasoning. A separate problem is state stability: as an agent accumulates claims, hypotheses, and proof obligations, local updates may create global inconsistency. Verified-agent frameworks such as SEVerA impose formal constraints on generated outputs (Banerjee et al., 2026); our work addresses the complementary substrate problem of certifying global coherence of the evolving state graph itself.

3. Agent-State Model and System

We use "agent state" structurally. Vertices represent claims, hypotheses, proof obligations, evidence nodes, domain concepts, or intermediate scientific objects. Edges represent compatibility, dependency, support, or restriction relations. The sheaf records local data and transition constraints. A proposed self-modification changes this graph-sheaf pair; the verification problem is whether the modified state remains globally coherent.

Semantic loader interface and coherence. Populating the graph-sheaf pair requires a *semantic loader*: a front-end mapping claims to vertices, relations to edges, and inter-claim constraints to restriction maps. Algebraically, the substrate is loader-agnostic; empirically, its utility is bounded by loader fidelity. The verifier certifies structural coherence of the represented state, not semantic truth. A false statement may register zero coboundary if represented compatibly with its neighbors; a true statement may be rejected if mapped inconsistently. Truth-sensitive verification requires that axioms, proof obligations, or domain semantics be encoded as sheaf constraints. Here "contradiction" means a represented structural incompatibility (local gluing failure, unsupported restriction), broader than syntactic $P \wedge \neg P$ but narrower than semantic falsity.

3.1. Sheaf Substrate

Let $G(t) = (V(t), E(t))$ be a finite graph at discrete time t with $r(t) = |E(t)|/|V(t)|$. A cellular sheaf $\mathcal{F}(t)$ as-

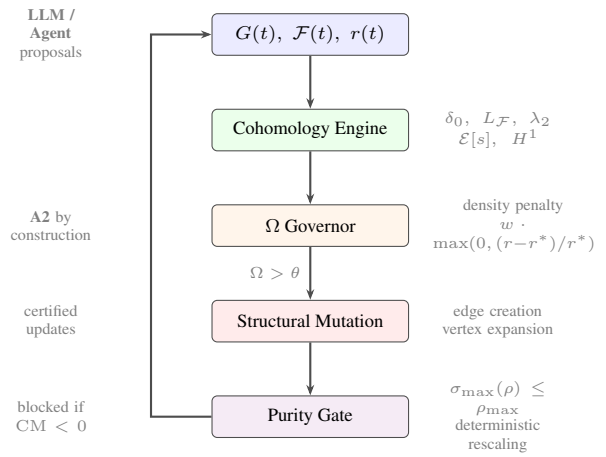

Figure 1. Verification loop: proposal → sheaf verification (obstruction, energy) → failure localization (H^1, Dirichlet) → commit or intervene. (A2 by construction: the governor supplies monotone restorative drift, §3.3.)

signs stalk spaces $\mathcal{F}(v) = \mathbb{R}^d$ ($d = 8$ in all experiments) to vertices and restriction maps $\rho_{u\to e} : \mathcal{F}(u) \to \mathcal{F}(e)$ to incidences. A stalk encodes the local content of a claim (e.g., a qualifier feature vector), and a restriction map encodes the constraint adjacent claims must satisfy: if vertex u represents "all bounded sequences have convergent subsequences" and vertex v drops "bounded," the restriction maps produce non-zero coboundary on the shared edge, flagging the incompatibility.

The coboundary operator $\delta_0 : C^0 \to C^1$:

$$(\delta_0 s)(e) = \rho_{v\to e}(s(v)) - \rho_{u\to e}(s(u)) \tag{1}$$

The sheaf Laplacian $L_{\mathcal{F}} = \delta_0^\top \delta_0$ (positive semidefinite). Dirichlet energy $\mathcal{E}[s] = s^\top L_{\mathcal{F}} s$ measures restriction map disagreement. The spectral gap $\lambda_2(L_{\mathcal{F}})$ is the smallest positive eigenvalue, i.e., $\lambda_{d\cdot\beta_0+1}$ where β_0 is the number of connected components, excluding the $d\cdot\beta_0$ zero eigenvalues from constant sections. For a graph (1-dimensional CW complex):

$$\dim H^1(G;\mathcal{F}) \le d\cdot\beta_1, \quad \beta_1 = |E| - |V| + \beta_0 \tag{2}$$

Crucially, β_1 counts topological cycle capacity but is blind to algebraic compatibility: a cycle whose restriction maps compose to the identity contributes zero to $\dim H^1$, while the same cycle with incompatible maps contributes a nonzero obstruction. Sheaf cohomology is strictly more informative than graph-theoretic Betti numbers; restriction maps are the minimal structure needed to distinguish consistent cycles from contradictory ones.

3.2. Purity Gate (Bounded-Gain Certification)

All restriction maps satisfy a bounded spectral-radius condition enforced at construction:

$$\text{(B3)} \quad \sigma_{\max}(\rho) \le \rho_{\max} < 1 \tag{3}$$

Any map exceeding the bound is uniformly rescaled before admission. This preserves the map direction while guaranteeing strict contractivity. The bound is an architectural invariant, not a post-hoc diagnostic. Stability arguments depend only on $\rho_{\max} < 1$, not on initialization scale or construction distribution. This places the sheaf in the near-identity regime of First & Kaufman (2024).

Semantic interpretation and expressivity cost. The contractivity bound applies to the verifier, not to scientific inference: it limits how much a single incidence map can amplify disagreement. Expansive mathematical moves are admitted only after factorization into bounded proof-state steps: intermediate lemmas, subgoals, coercion/typeclass nodes, abstraction nodes, or evidence nodes whose restriction maps each pass the gate. The cost is a longer represented path and more vertices, not a smaller reasoning horizon. This topological friction is exposed by the 990-lemma study: coarse natural-language loaders often fail to supply the needed factorization, whereas formal kernels such as Lean already decompose reasoning into bounded steps.

3.3. Dynamics and Density-Aware Governor

Two competing processes modify $G(t)$ at each step. Edge creation adds edges at a bounded rate subject to saturation and duplicate rejection. Vertex expansion is triggered by a pressure functional $\Omega(t)$ aggregating per-vertex Dirichlet energy $\mathcal{E}[s]$ and the spectral obstruction proxy $\mu(t)$ (excess near-zero eigenvalues of $L_{\mathcal{F}}$). When $\Omega(t)$ exceeds threshold θ, the governor requests a decomposition vertex: an intermediate lemma, subgoal, missing-hypothesis node, or evidence node that factors a high-pressure relation into smaller certified constraints. In the experiments this vertex is instantiated structurally to isolate the stability mechanism; in deployed proof-state systems it would be supplied by the planner or proof kernel, and all new restriction maps pass through the Purity Gate. The governor is not a learned policy but an architectural mechanism supplying the monotone restorative drift of Assumption A2 by construction. The pressure functional:

$$\Omega(t) = \sum_{v\in V} \mathcal{E}[s](v) + \alpha\cdot\mu(t) + w\cdot\max\Big(0,\ \tfrac{r(t)-r^*}{r^*}\Big) \tag{4}$$

where $\mathcal{E}[s](v)$ is local Dirichlet energy and $\mu(t)$ is the spectral obstruction proxy (excess near-zero Laplacian eigenvalues beyond β_0).

Edge proposal, duplicate rejection, and certified restriction-map admission are in Supplement D, Algorithm 1.

The Purity Gate provides intrinsic stability by bounding edgewise error amplification; the density term in Ω supplies the monotone pressure needed for A2 in the governed regime. The density-aware governor provides extrinsic stability. Here w is a tunable weight and r^* the target attractor, measured empirically near 3.0 for the governed dynamics studied here. When E/V exceeds r^*, the density term pushes Ω above the warning threshold, triggering vertex expansion that dilutes E/V back toward r^*. This gives $b'(r) = w/r^*$ for $r > r^*$, satisfying A2 by construction.

3.4. Spectral Obstruction Proxy

The implementation tracks a spectral proxy $\mu(t)$ (excess near-zero eigenvalues beyond β_0) rather than exact $\dim H^1$. On the canonical trajectory, $\mu(t)$ correlates with exact $\dim H^1$ at $r = 0.89$ ($p < 10^{-4}$). Stability arguments rely only on exact β_1 and the Laplacian spectral gap; the proxy serves solely as a diagnostic trigger for the governor (Supplement H).

3.5. Cellular Decomposition

We use "cell" here to refer to a bounded subgraph partition for streaming Mayer–Vietoris assembly, distinct from the topological cells of the underlying CW complex.

Monolithic eigensolve and cohomological recomputation scale as $O(n^3)$. We bypass this via a bounded-cell partition: $G(t)$ is decomposed into cells of strictly bounded size, localizing all eigensolves to constant-time subproblems. Global H^1 is assembled from local cell contributions via a Mayer–Vietoris sequence. Under streaming edge insertions, only the affected cell requires recomputation; all others remain cached, reducing amortized per-edit cost from $O(n^3)$ to $O(1)$ in global graph size (Figure 2). Cells exceeding a fixed size bound are split via greedy bisection, incurring occasional $O(K)$ rebalancing where K is the cell-size cap (Supplement J). The $O(1)$ claim is with respect to global graph size for fixed cell-size and boundary caps; it does not cover arbitrary repartitioning or unbounded boundary growth.

4. Theorems and Assumptions

The implemented system is fully deterministic given a seed. We analyze the closed-loop dynamics through the stochastic approximation (SA) framework of Borkar (2008). The perturbation terms $\xi(t)$ and $\xi_\lambda(t)$ are bounded deterministic disturbances; the step-size $1/n(t)$ and drift conditions satisfy the Borkar tracking hypotheses regardless of stochastic interpretation. Equivalently, this is a deterministic Lyapunov analysis with $O(1/n)$ discretization error; we cite Borkar as the analytic tool, not as a claim about randomness. The perturbation terms represent the deterministic but externally

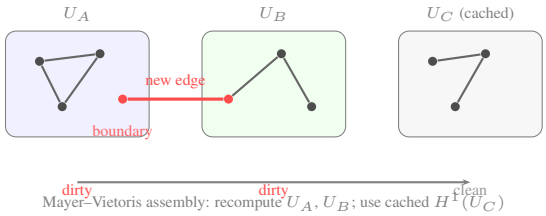

Figure 2. Cellular decomposition under streaming edge insertion. A new edge between U_A and U_B (red) marks both dirty; U_C remains cached. Global H^1 is assembled from recomputed local cohomology of dirty cells and cached values of clean cells via Mayer–Vietoris. Per-edit cost is $O(1)$ in $|V|$.

Table 1. Claim taxonomy and discharge. The verifier is exact relative to the represented sheaf; stability and streaming cost are conditional on stated assumptions.

Claim	Status	Discharge
Coboundary, energy, assembled H^1	Exact	Deterministic linear algebra on the represented sheaf; ProofDAG validates exact separation.
Purity Gate $\sigma_{\max} \leq \rho_{\max} < 1$	Architectural invariant	Enforced by deterministic rescaling before map admission.
Density stability (E/V)	Policy-conditional	A1–A4; A2 supplied by governor, A1 by admission control, A3–A4 measured.
Spectral-gap stability	Policy-conditional	B1–B4; B3 enforced by Purity Gate, B1/B2/B4 empirical.
$O(1)$ per-edit maintenance	Algorithmic conditional	Bounded cell size, bounded boundary, cached Mayer–Vietoris; validated 21K–5M vertices.
NL math verification	Representation-limited	990-lemma study measures loader aliasing, not substrate failure.

driven arrival of agent proposals, not internal stochasticity. We use Borkar SA because the natural step size is $1/|V(t)|$: each edit changes a vanishing fraction of the global state, matching the tracking-ODE regime.

We distinguish three categories of claims: (i) architectural invariants enforced by construction (e.g., $\sigma_{\max}(\rho) \leq \rho_{\max} < 1$ via the Purity Gate), (ii) conditional theoretical results under explicitly stated assumptions, and (iii) empirical observations on the implemented system (e.g., r^*, λ_2^*, and the restorative drift condition B1). The algebraic verifier is exact relative to the represented sheaf: coboundary, Dirichlet energy, and assembled cohomology are deterministic structural tests. The long-run stability claims are narrower: they require the governed update policy to satisfy A1–A4 and B1–B4, and the measured constants ($r^* \approx 3.0$, $\lambda_2^* \approx 0.530$) are properties of the implemented regime, not universal sheaf constants. Key assumptions are enforced by construction (A2 by the governor, B3 by the Purity Gate); violating empirical conditions (B1, B2) invalidates stability theorems but does not disable update-time coherence checking. The results are a systems theorem for the implemented governed substrate, not a universal theorem for arbitrary sheaves.

4.1. Theorem 1 (Structural Density Stability)

Theorem 1 (Structural Density Stability)**.** *Assume bounded edge rate (A1: saturation gives $\bar{m}'(r) \leq 0$ near r^*), monotone expansion frequency (A2: the governor gives $b'(r) > 0$), unique equilibrium (A3: measured $r^* = \bar{m}/\bar{n} \approx 3.0$), and small single-step updates (A4: $|\Delta r| = O(1/n)$, $C_m = 12$). Then $r(t)$ converges to an $O(1/n)$-neighborhood of r^*. The theorem is a systems guarantee: the control logic of Equation 4 counteracts the bounded edge rate without overshoot or oscillation.*

Proof sketch. The Lyapunov function $V(r) = (r - r^*)^2$ has negative drift away from r^* by A1–A2; Borkar tracking (Borkar, 2008) yields the stated neighborhood. Full proof in Supplement A.

4.2. Theorem 2 (Policy-Conditional Spectral Gap Stability)

Theorem 2 (Spectral Gap Stability)**.** *Assume restorative spectral drift (B1: $h'_\lambda(\lambda_2^*) < 0$, supported empirically, slope $= -75.31$, $p = 0.002$), empirical spectral perturbation rate (B2: $|\Delta\lambda_2| \leq C_\lambda/n$, $C_\lambda = 73.21$), Purity Gate regularity (B3: $\sigma_{\max}(\rho) \leq \rho_{\max} < 1$, enforced by construction), and stationary positive spectral equilibrium (B4: $\lambda_2^* > 0$, ADF $p = 0.016$). Then $\lambda_2(L_{\mathcal{F}}(t))$ converges to an $O(1/n)$-neighborhood of $\lambda_2^* > 0$.*

Proof sketch. The Lyapunov function $W(\lambda_2) = (\lambda_2 - \lambda_2^*)^2$ has negative drift away from λ_2^* by B1; Borkar tracking yields the stated neighborhood. Full proof in Supplement B.

The per-edge Laplacian bound $\|L_e\|_{\text{op}} \leq 2\rho_{\max}^2$ and rank $\leq 2d$ are rigorous (Lemma 1, Supplement C); the $O(1/n)$ displacement rate relies on empirical local spectral-density regularity specific to the implemented mutation policy. Theorem 2's convergence rate is conditional on this observed regularity, not a universal law for arbitrary growing sheaves. The theorem is a closed-loop systems guarantee: the algebraic verifier is exact at each update, while long-run stability follows only for policies satisfying the stated drift and bounded-perturbation conditions.

Observation (Empirical Restorative Spectral Drift Under the Governed Policy). *This observation supports B1 empirically, not as a universal derivation.* Under A1–A4, B3, and the empirical condition that spectral-gap deviations are monotonically coupled to density excursions in the governed regime, $h'_\lambda(\lambda_2^*) < 0$. Vertex expansion dilutes the spectral gap (Cauchy interlacing), certified edge additions have bounded perturbation (Lemma 1), and the governor supplies restorative feedback. The measured slope $h'_\lambda = -75.3$ ($p = 0.002$) supports B1 for the implemented policy.

Table 2. Comparison of the two stability results ($L_{\mathcal{F}}$: sheaf Laplacian of $\mathcal{F}$).

	Theorem 1	Theorem 2
Subject	E/V	$\lambda_2(L_{\mathcal{F}})$
Equilibrium	$r^* \approx 3.0$	$\lambda_2^* = 0.530$
Noise bound	$C_m = 12$	$C_\lambda = 73.21$
Drift slope $\lvert h' \rvert$	$\sim$1.2	$\sim$75.3
Attractor width	$\sim$0.025	$\sim$0.005
$O(1/n)$ source	Algebraic	Spectral (rank)

4.3. Restricted Stability Corollary for Certified Sheaves

Corollary 1. *Under A1–A4 and B1–B4, certified dynamical sheaves simultaneously satisfy: (i) $r(t) \to r^*$ with $O(1/n)$ width; (ii) $\lambda_2(L_{\mathcal{F}}(t)) \to \lambda_2^* > 0$; (iii) $\beta_1/n \to r^* - 1$; (iv) $\dim H^1 \leq d \cdot \beta_1$; (v) the tracked spectral gap remains bounded away from zero in the connected governed regime.*

This is a restricted result for certified bounded-gain sheaves, not a universal sheaf Cheeger theorem. First & Kaufman (2024) show no universal Cheeger inequality extends to arbitrary sheaves; our result applies only to the closed-loop regime with Purity Gate, graph connectivity, and density-aware governor. The equilibria r^* and λ_2^* are system-specific; an update policy violating these constraints invalidates the guarantees.

4.4. Coherence Margin

The coupled stability of r and λ_2 provides the formal basis for the Coherence Margin (CM). For a proposed update supported on a local cell or neighborhood U, CM is the signed margin

$$\text{CM}(U,t) = \gamma_\lambda \lambda_2(L_{\mathcal{F}|U}) - \Big(\textstyle\sum_{e\in U} \|(\delta_0 s)(e)\|^2 + \alpha\, \mu_U(t)\Big), \tag{5}$$

where the first term is local spectral resolution capacity and the second is contradiction pressure from coboundary energy and excess near-zero obstruction modes; γ_λ and α are unit-free scaling weights (order 1 and 0.1 respectively in all experiments). If $\text{CM}(U,t) < 0$, the update is blocked or routed to decomposition; otherwise it may proceed through the Purity Gate and bounded-cell update rules. The global CM variant (Supplement K) uses EMA-smoothed resolution and creation rates for operational monitoring.

5. Empirical Validation

5.1. ProofDAG: A Formal Proof Dependency Corpus

We construct ProofDAG, a synthetic proof dependency corpus serving as a controlled unit test for exact-representation coherence, validating the algebraic and spectral machinery with controlled contradiction injection and ground-truth labels.

Table 3. Coboundary norm separation on ProofDAG (seed 42). Clean edges have exactly zero coboundary; obstruction edges produce norms in $[0.51, 1.25]$. Gap: 0.505.

	Count	Mean	Min	Max
Clean edges	2330	0.000	0.000	0.000
Axiom conflict	32	1.009	0.505	1.255
Circular	16	0.945	0.709	1.234
Type mismatch	18	0.886	0.584	1.088
All obstructions	66	0.960	0.505	1.255

Table 4. Multi-seed validation on ProofDAG. All seeds achieve F1 $= 1.000$ with zero false positives. E/V ratio: mean $= 2.402$, CV $= 0.17\%$.

Seed	$\lvert V\rvert$	$\lvert E\rvert$	E/V	F1	Gap	Cells
42	999	2396	2.40	1.000	0.505	54
72	1000	2409	2.41	1.000	0.311	60
137	1000	2402	2.40	1.000	0.528	58
271	999	2396	2.40	1.000	0.583	56
Mean / CV			2.40	1.000	0.482	CV $= 0.17\%$

ProofDAG models a growing proof library across 10 mathematical domains mirroring mathlib4: 1000 vertices, approximately 2330 clean dependency edges. The deterministic loader constructs restriction maps as 0.68-scaled random orthonormal matrices (QR of Gaussian, then uniform rescaling), ensuring clean dependencies are global sections ($\delta_0 s = 0$) and all maps pass the Purity Gate ($\sigma_{\max} \leq 0.95$). Contradictions are injected as controlled mutations (66 total): circular dependencies, axiom conflicts, and type mismatches, ensuring non-zero coboundary with full ground-truth labels.

5.2. Coherence-Obstruction Detection via Coboundary Norm

The detection mechanism is a direct algebraic test: for each edge (u, v) with restriction maps (ρ_u, ρ_v) and stalk vectors $(s(u), s(v))$, compute

$$c(e) = \|\rho_{v\to e}(s(v)) - \rho_{u\to e}(s(u))\| \tag{6}$$

For clean dependencies, $c(e) = 0$ by construction. For obstruction-labeled edges, $c(e) > 0$ with mean 0.96 across all seeds. These labels denote represented compatibility failures, not arbitrary semantic falsity. An edge is flagged if $c(e)$ exceeds $\tau = \mu_{\text{clean}} + 3\sigma_{\text{clean}}$. Since all clean edges have $c(e) = 0$ exactly, $\tau = 0.01$ (safety floor), and the minimum obstruction norm is 0.31, the distributions do not overlap (Table 3).

5.3. Detection Accuracy

The per-edge coboundary test achieves exact separation across all four seeds (Table 4): precision $= 1.000$, recall $= 1.000$, F1 $= 1.000$, zero false positives and zero false negatives. This is a unit test for both the algebra and streaming infrastructure: exact separation confirms that cellular decomposition and Mayer–Vietoris assembly do not leak algebraic drift. The minimum obstruction norm (0.31 across seeds) exceeds the maximum clean norm by a factor of 31.

5.4. Streaming Dynamics on Proof Graphs

Proof dependency graphs are streamed through the cellular decomposition in topological order, exercising the full pipeline. All 2396 edges are ingested via the StreamingBuilder with case classification (Case A: intra-cell, Case B: cross-cell, Case C: new vertex, Case D: seed cell). The decomposition produces 54–60 cells depending on seed, with median per-edge latency of 50.8 μs. The E/V ratio remains stable at 2.40 ± 0.004 (CV $= 0.17\%$) across all seeds, showing bounded-density behavior on externally structured proof graphs. ProofDAG's lower ratio (≈ 2.4) reflects its tree-like structure; the governor is inactive here, so this run validates streaming infrastructure rather than Theorem 1's governed attractor. We next test this exact-loader path on a small real Lean/mathlib subset.

Lean exact-loader benchmark. To validate the exact-loader path on a real formal library, we ran a 6-namespace mathlib benchmark covering arithmetic, algebra, order theory, lists, integers, and logic: 507 declarations and 954 kernel-certified dependency edges (Supplement S). The loader consumes Lean's elaborated environment, deterministically encoding each declaration's role, dependency identity, hypothesis footprint, and typeclass/coercion footprint into an $\mathbb{R}^8$ feature vector; restriction maps are instantiated from this typed dependency signature and passed through the Purity Gate. All 954 clean dependencies produced zero coboundary up to tolerance (max residual 2.61×10^{-16}), while 62 loader-level controlled corruptions covering hypothesis removal, condition dropping, wrong dependency, and circular dependency were detected by nonzero coboundary (minimum norm 0.514). All sampled restriction maps passed the Purity Gate ($\max \sigma_{\max} = 0.760 < 0.95$). This benchmark validates that a proof kernel can supply the exact representation abstracted by ProofDAG across diverse mathematical domains (per-namespace breakdown in Supplement S). A commit-gate case study (Supplement S) further tests the substrate as a pre-commit gate: corrupted updates (wrong dependency, condition dropping, circular dependency) produce negative CM and are blocked with fault localization to the corrupt edge; targeted repairs (intermediate lemma insertion, missing hypothesis restoration, bridge declaration) restore

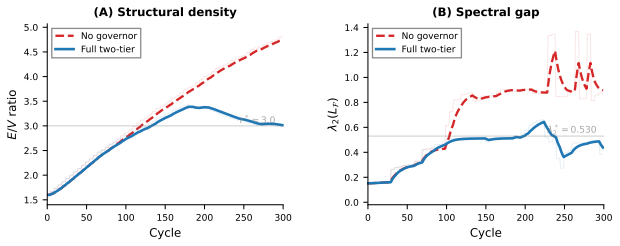

Figure 3. Governor ablation (seed 42, 300 cycles). (A) Without the governor, E/V drifts to 4.8; the full architecture constrains density to $r^* \approx 3.0$. (B) Uncontrolled density drives λ_2 above 0.8; the governed system stabilizes at $\lambda_2^* = 0.530$.

zero coboundary and nonnegative CM, allowing commit.

5.5. Live LLM Dependency Hallucination Audit

To test the "neural proposes, sheaf verifies" loop on real LLM output, we prompted a frontier language model to identify the direct dependencies of all 381 theorems across all 6 namespace groups. The model claimed 1,046 dependencies total, of which only 314 (30.0%) matched the kernel-certified ground truth; the remaining 732 were hallucinated. The sheaf substrate detected all 732 hallucinated dependencies via nonzero coboundary (min norm 0.292, mean 0.934), with fault localization to the exact hallucinated edge, using zero learned parameters. All 314 correct dependencies produced zero coboundary (max residual 2.61×10^{-16}). The detection rate was 100% across all 6 namespace groups individually (full per-theorem breakdown in Supplement S). This audits proposed-edge validity, not whether the model recovered every true dependency.

5.6. Attractor Behavior on Governed Dynamics

The primary prediction of Theorem 1 is convergence of $r(t) = |E|/|V|$ toward a narrow attractor. In the canonical 500-cycle run (seed 42, 150-vertex initial graph), the system enters a stable regime near $r^* \approx 3.0$: mean $r = 3.0$, std 0.02, range $[2.9, 3.1]$ over the 350-cycle attractor window. Cross-seed validation (seeds 42, 72, 137, 271) over 300 governed cycles yields final ratios $[3.0, 3.0, 3.1, 3.1]$ with CV $= 1.9\%$. The bounded edge rate (A1) acts as architectural admission control, ensuring the governor's restorative drift is never overwhelmed.

5.7. Scale Validation

The cellular decomposition enables operation at scales beyond the stability validation runs. Table 5 summarizes streaming performance across four orders of magnitude.

Sublinear scaling (exponent 0.19, $R^2 = 0.975$ across $V = 21$K to 1M) confirms cost is bounded by local cell size, not graph size. Memory scales linearly with cell count;

Table 5. Incremental verification at scale (single machine, commodity hardware; methodology, hardware specs, and measurement protocol in Supplement J). "Drift" is algebraic invariance of assembled cohomology dimension (integer-valued), verified by comparing incremental Mayer–Vietoris assembly against monolithic recomputation at checkpoints.

V	Cells	Edit (med)	Edit (mean)	Drift	
21K	639	31 μs	31 μs	0	
100K	890	46 μs	46 μs	0	Median
1M	8,663	63 μs	63 μs	0	
5M	25,473	35 μs	119 μs	0	

reported for 5M; mean is higher due to $\sim$15% cross-cell boundary insertions triggering two local recomputes (Supplement J).

at V=5M, map deduplication reduces the RestrictionStore to 0.50 MB for the implemented map pool, and full cached state including stalks, local cohomology, and boundary terms fits in commodity RAM (Supplement J). The 5M result demonstrates that verification infrastructure is not the bottleneck; loader fidelity (§5.9) is the binding constraint on end-to-end utility.

5.8. Robustness to Loader Noise

In deployed systems, LLM-assisted loaders will introduce noise. We inject isotropic Gaussian noise $\mathcal{N}(0, \sigma^2)$ into both restriction maps of every clean ProofDAG edge, reenforce the Purity Gate, and leave contradiction maps fixed. Rescaling is applied after noise injection, so any compression of the coboundary signal by the contractivity bound is already reflected in reported rates. Up to $\sigma = 0.10$ (14% relative noise), the system maintains F1 $= 0.948$ (precision $= 0.928$, recall $= 0.970$). Degradation beyond 20% is driven by false negatives (noise overwhelms the obstruction signal), not false positives. The substrate's guarantee remains conditional on loader fidelity, but this ablation demonstrates significant noise tolerance (full table in Supplement Q). This tests numerical resilience, not tolerance to structural hallucinations (§5.9).

5.9. Structural Aliasing at Scale (990-Lemma LLM Study)

This experiment tests whether a coarse natural-language set-lattice loader preserves enough mathematical structure for autonomous commit decisions. It does not; the loader aliases valid generalization and invalid condition dropping, identifying representation fidelity as the binding constraint.

To close the "neural proposes, sheaf verifies" loop with real LLM outputs, we conduct a 990-lemma study: a frontier language model generates candidate lemmas across 10 mathematical domains (490 valid generalizations, 500 plausible-sounding statements that drop or weaken conditions from

Table 6. Representation-fidelity ladder. Detection is exact when the representation is exact, including a 6-namespace Lean/mathlib benchmark and a live LLM dependency audit (732/732 hallucinations caught); it degrades gracefully under noise and fails under coarse natural-language loaders that erase mathematical roles. The 990-lemma result is a loader-fidelity bound, not a cohomological-maintenance bound.

	Prec.	Rec.	F1	Acc.
ProofDAG (exact)	1.000	1.000	1.000	1.000
Lean/mathlib (6 ns groups)	1.000	1.000	1.000	1.000
LLM proposed-edge audit (381 thms)	1.000	1.000	1.000	1.000
ProofDAG + 14% noise	0.928	0.970	0.948	–
ContractNLI (coboundary only)	0.000	0.000	0.000	0.500
Set lattice (20 lemmas)	0.588	1.000	0.741	0.650
Set lattice (990 lemmas)	0.502	0.756	0.603	0.498

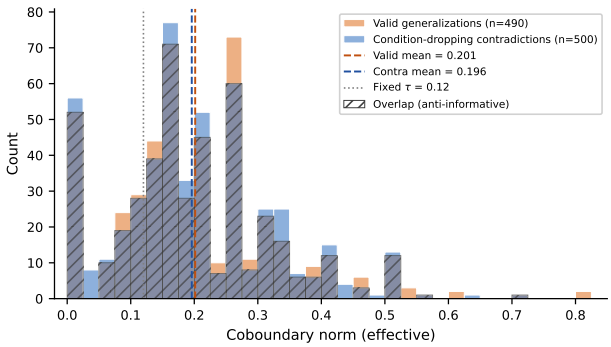

Figure 4. Coboundary norm distributions for 490 valid generalizations (orange) and 500 condition-dropping contradictions (blue). Distributions overlap almost completely; the valid mean (0.201) exceeds the contradictory mean (0.196), rendering the set-lattice loader anti-informative for autonomous commit decisions. No threshold separates valid generalization from invalid condition dropping.

established theorems, constituting represented structural contradictions per §3.1). The LLM receives domain axioms and theorems as context; ground truth is controlled by prompting valid and contradictory batches separately.

Each lemma is evaluated through a deterministic zero-ML set-lattice loader (Supplement R) that assigns coboundary scores from structural overlap and missing qualifiers. Table 6 reports results at fixed threshold $\tau = 0.12$.

Structural aliasing at scale. The 990-lemma study confirms that the aliasing observed in the 20-lemma pilot is a fundamental limitation of set-difference semantics, not a small-sample artifact. Valid and contradictory coboundary distributions overlap almost completely (means 0.201 vs. 0.196; Figure 4), rendering the signal effectively anti-informative (valid scores exceed contradictory scores). At $\tau = 0.12$, recall is 0.756 and precision 0.502. The 375 false positives and 122 false negatives expose a representational ceiling, not a tuning failure.

Implications for autonomous verification. The error profile is asymmetric: false negatives corrupt state and false positives block valid evolution, so at precision 0.502 the set-lattice loader is unsuitable for autonomous commit decisions. It remains useful as a conservative safety filter, surfacing 75.6% of condition-dropping failures for escalation while governed dynamics maintained $r(t) \in [2.95, 3.05]$ and $\lambda_2 \approx 0.53$, consistent with Theorems 1–2. Even on strict DAGs where H^1 vanishes, per-edge coboundary norm and Dirichlet energy remain informative for detecting local incompatibilities, as demonstrated by the Lean benchmark and hallucination audit above (further discussion in Supplement S).

6. Discussion, Limitations, and Conclusion

We present a policy-conditional cellular-sheaf substrate for update-time coherence checking of evolving scientific-agent state, not a universal stability theorem for arbitrary agents or sheaves. The substrate generalizes structurally (ProofDAG), tolerates approximate representation (noise ablation), and exposes the natural-language loader boundary (990-lemma study). Figure 3 confirms that stability is a property of the enforced closed-loop regime: removing the governor breaks both attractors.

Vertex expansion semantics. Governor-triggered expansions are decomposition requests, not semantically empty proof objects. Lean-style kernels are natural exact loaders for this regime: they certify each local typed object, while the sheaf layer supplies global structural telemetry over the evolving dependency state. Our experiments instantiate these vertices structurally to isolate the stability mechanism; full planner- or kernel-generated decomposition content is future work. As quantified in §5.9, current natural-language loaders cannot perform the required compositional decomposition, motivating richer algebraic signatures (e.g., Heyting-algebraic loaders) for future mathematical loaders.

This system is an update-time structural verifier: exact for represented structure, conditional for governed dynamics, and empirical for loader fidelity and scale. Deletions and retractions are handled symmetrically by the streaming infrastructure but fall outside the current Lyapunov guarantees (Supplement J).

Impact Statement

This work provides a structural verification layer for AI systems that accumulate scientific knowledge, enabling update-time detection of coherence failures before commit. Operational risks, including false negatives and false positives, are quantified empirically. The system is a verifier, not a generator.

References

Ayzenberg, A., Gebhart, T., Magai, G., and Solomadin, G. Sheaf theory: From deep geometry to deep learning. *arXiv preprint arXiv:2502.15476*, 2025.

Bamberger, J. et al. Bundle neural networks for message diffusion on graphs. In *International Conference on Learning Representations (ICLR)*, 2025. Spotlight. arXiv:2405.15540.

Bandeira, A. S., Singer, A., and Spielman, D. A. A Cheeger inequality for the graph connection Laplacian. *SIAM Journal on Matrix Analysis and Applications*, 34(4):1611–1630, 2013.

Banerjee, D., Xu, C., and Singh, G. SEVerA: Verified synthesis of self-evolving agents. *arXiv preprint arXiv:2603.25111*, 2026.

Barabási, A.-L. and Albert, R. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

Barbero, F. et al. Sheaf neural networks with connection Laplacians. In *ICML TAG-ML Workshop*, 2022.

Bodnar, C. et al. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in GNNs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Borkar, V. S. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

Borkar, V. S. Stochastic approximation in non-Markovian environments revisited. *arXiv preprint arXiv:2603.21091*, 2026.

Dalrymple, D. et al. Towards guaranteed safe AI. *arXiv preprint arXiv:2405.06624*, 2024.

First, U. and Kaufman, T. Cosystolic expansion of sheaves on posets. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC)*, 2024.

Gross, T. and Blasius, B. Adaptive coevolutionary networks: A review. *Journal of the Royal Society Interface*, 5(20): 259–271, 2008.

Hansen, J. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3:315–358, 2019.

Hansen, J. and Gebhart, T. Sheaf neural networks. *arXiv preprint arXiv:2012.06333*, 2020.

Hernandez Caralt, F. On the necessity of learnable sheaf Laplacians. *arXiv preprint arXiv:2603.05395*, 2026. GRaM Workshop at ICLR 2026.

Hu, C.-S. et al. A sheaf-theoretic and topological perspective on complex network modeling and attention mechanisms in graph neural models. *arXiv preprint arXiv:2601.21207*, 2026.

Koreeda, Y. and Manning, C. D. ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 1907–1919, 2021.

Le Vu Anh, D. et al. Lipschitz bounds for persistent Laplacian eigenvalues. *arXiv preprint arXiv:2506.21352*, 2025.

Revay, M. et al. Lyapunov-stable deep equilibrium models. In *AAAI Conference on Artificial Intelligence*, 2024.

Richards, S. M. et al. The Lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems. In *Conference on Robot Learning (CoRL)*, 2018.

Wei, X. and Wei, G.-W. Persistent sheaf Laplacians. *Foundations of Data Science*, 7(2):221–255, 2025. First circulated 2021, published 2025.