

The Constraint Theory of Autonomous Agency

Self-Indexing Manifolds and the Boundaries of Unified Systems

Nova Spivack

March 2026

Preprint. Not peer-reviewed.

Archival version (Zenodo): [DOI pending publication]

Abstract

We formalize the Self-Indexing Adjudicative Manifold (SIAM) as a bounded dynamical regime in phase space—a specialized unified-agent regime inside an already-proved closure/selection/diagonal framework (NEMS, viable continuation). This paper proves a bounded dynamical regime for a specialized unified-agent class inside the closure/selection framework; it does not claim that all sentience reduces to O-SIAM alone. O-SIAM is defined via seven structural invariants with witness-carrying structures: refining ledger, self/other partition, recursive self-update, mirror (coverage, freshness, non-exhaustion), adjudication, reconciliation, and encoding robustness. We establish timescales, Master Bottleneck Λ , openness band, and topology on *ReconciliationSimplex*. Pathologies map to Paper 71 [10] defects via explicit embedding ι and `toVCDefectProfile`. All load-bearing flagship theorems are machine-checked (summit, mirror non-exhaustion from Paper 51, burden-VC capacity, ridge pathology mapping, separation). DSAC validation via `sentience-compute` (64+ runs, 10 scenarios, TDA and burden on all). P-SIAM (phenomenological extension) is demoted to appendix. This overview presents the core NEMS theorem engine and selected applications; stronger domain-specific derivation and ontological synthesis claims belong to separate release surfaces with their own premise bundles and formal artifacts.

Trust boundary. DSAC and TDA outputs are *computational consistency checks*; the certificate layer is **sentience-lean** [9]. Interpretive bridges to phenomenology (P-SIAM) or consciousness are explicitly scoped; see Section 1 and Section A.

Archival version (Zenodo): [DOI pending publication]

Keywords: self-indexing manifold, unified agency, viability boundary, topological data analysis, formal verification

MSC 2020: 03B70, 68Q85, 92B20

Code/Data availability: Lean 4 **sentience-lean** [9] — `lake build` at repository root; theorem names in main text and `MANIFEST.md`.

Computational validation: `sentience-compute`; engine `delta-machine`; see `sentience-compute/README.md`.

Verification steps: Section A.

Paper 73 (sentience flagship).

Contents

1	Premises, disagreement coordinates, and trust boundary	4
2	How formal claims map to verification (human layer)	4
3	The Crisis of Agency	4
4	The Operational SIAM (O-SIAM)	5
4.1	Process Window and Viable Subject-Window	5
4.2	Seven Structural Invariants	5
5	The Mathematical Bounds	5
5.1	Why Information Geometry and Topology Are Natural for SIAM	6
5.2	Why the Topological Conditions Correspond to SIAM Invariants	6
5.3	Structural-to-Topological Bridge Theorems	6
5.4	Foothill Theorems	6
6	Information-Theoretic Burden	7
7	Separation Theorems	7
8	Phase Distinction	7
9	Pathologies of Agency	8
9.1	Ridge Theorems (Proved)	8
9.2	Summit Theorem	8
10	DSAC Empirical Validation	8
10.1	What Is DSAC and Why Is It Applicable?	8
10.2	Pipeline and Scenarios	9
10.3	Theorem-Relevant Analysis	9
10.4	Production Results (2026-03-08)	10
11	Discussion	10
11.1	Comparison with IIT, GWT, Active Inference, and Related Frameworks	10
11.2	The Bridge to Viable Continuation: A Theory of How Unified Systems Fail	11
11.3	Implications for AI	11
11.4	Implications for Neuroscience and Cognitive Science	11
11.5	Philosophical Implications	12
11.6	Operational Traction: Why the Computational Validation Matters	12
11.7	Summary: Boundary, Taxonomy, and the Fog Edges	12
12	What Paper 73 Does Not Claim	12
13	Tools Generated by the Theory	13

14 Lean Correspondence Table and What Is Defined vs Proved vs Simulated	13
14.1 What Is Defined	13
14.2 What Is Proved (Machine-Checked)	13
14.3 What Remains as Bridge Axioms (Explicit Provenance)	14
14.4 What Is Scaffold	14
14.5 What Is Simulated	14
15 P-SIAM (Extension/Appendix)	14
A Guide to machine-checked verification (sentience flagship)	14
B DSAC: Architecture, Applicability, and Validity	15
C Simulation-to-Theory Correspondence	15
D TDA Production Results	15
E Bridge Discharge Status	16
F Burden Decomposition and Validation Results	17
G Encoding Robustness and Cross-Architecture	17
H Geometric and Topological Atlas of the SIAM Process-Window	18

1 Premises, disagreement coordinates, and trust boundary

Where disagreement can (and cannot) live

Logical posture. Flagship claims cited as **proved**/machine-checked refer to **sentence-lean** [9]. Conditional on that artifact and its import graph (NEMS, Paper 51 mirror non-exhaustion, viable-continuation bridges where stated), the separation and summit theorems are audit targets in Lean.

Premise coordinates.

- **DSAC \neq proof.** Scenario runs may fail or disagree across hardware; they do not override Lean.
- **P-SIAM scope.** The phenomenological extension is appendix material; full formal phenomenology is Paper 74+, not re-proved here.
- **Implementation drift.** A reader on a different Lean/Mathlib pin may see build failure before judging prose outdated; align with the repository’s **lean-toolchain**.

Audit trail. Section [A](#); DSAC detail Section [B](#).

2 How formal claims map to verification (human layer)

Clone **sentence-lean**, run `lake build`. Search the main text for witness and theorem identifiers (e.g. `feedforward_not_OSIAM`). For DSAC replication, follow **sentence-compute/README.md** and correspondence notes Section [C](#).

3 The Crisis of Agency

AI and biology lack a formal boundary condition for unified, self-modeling systems. Feedforward controllers, stateful systems, and even self-modeling optimizers have been proposed as sufficient for agency or consciousness; none suffice. The gap is structural, not scalar.

A feedforward controller has no self-index: its outputs are deterministic functions of inputs, with no active boundary maintained by the system. A stateful system may carry hidden state, but statefulness alone does not imply a self-model or adjudicative openness. Self-modeling optimizers can maintain internal models, yet the mere presence of a model does not imply unified agency—the model may be fragmented, stale, or exhausted. Multi-agent systems distribute agency across parts; no single unified subject-window emerges. We prove strict separation: feedforward and stateful controllers do not imply OSIAM; robust SIAM implies a nontrivial unified regime (`feedforward_not_OSIAM`, `stateful_not_OSIAM`, `robust_SIAM_implies_unified`).

The need is for a *structural* characterization: exact temporal, topological, and informational bounds that any system must satisfy to maintain a unified, self-modeling, decision-making continuity-space. This paper provides that characterization, with machine-checked theorems and computational validation.

4 The Operational SIAM (O-SIAM)

The Operational SIAM (O-SIAM) is a purely structural, cybernetic, and information-theoretic object. It defines the boundary conditions for **unified agency**—the regime in which a system maintains a viable subject-window over time. All core math and bounds are proven here; phenomenology is a separate extension.

4.1 Process Window and Viable Subject-Window

A **ProcessWindow** σ is a temporally extended window $[t_0, t_1]$ with $t_0 < t_1$, a state trajectory, a record layer (NEMS-compatible), and timescales τ_W, τ_C , etc. The record layer carries semantic/admissibility structure compatible with NEMS observational language and world-types. The boundary L is not drawn by an external observer; it is the Markov Blanket / Information Bottleneck actively maintained by the system.

4.2 Seven Structural Invariants

O-SIAM requires seven invariants, each with **witness-carrying structures**, not bare propositions:

1. **Refining Ledger:** Filtration, refinement relation, arrow-of-time structure.
RefiningLedgerWitness.
2. **Self/Other Partition:** Partition, update rule, stability bound over window.
SelfOtherPartitionWitness. Failure yields dissociation (local-global decoupling).
3. **Recursive Self-Update:** Future self-index depends nontrivially on present via admissible update. Self-indexed unity.
4. **Mirror (split):** Coverage, Freshness, Non-Exhaustion.
Witnesses: MirrorCoverageWitness, MirrorFreshnessWitness, MirrorNonExhaustionWitness.
Non-Exhaustion proved from Paper 51 [5].
5. **Adjudication:** Live alternatives, internal resolution, commitment deposition.
AdjudicationWitness.
6. **Reconciliation:** Synchronization structure, latency bound.
ReconciliationWitness. Failure \Rightarrow fragmentation ($\beta_0 > 1$).
7. **Encoding Robustness:** Semantic preservation under admissible re-encodings.
encodingRobustness_preservation; EncodingRobustnessWitness.

OSIAM $\sigma := \text{Nonempty OSIAMWitness}$ —a unified agency regime. All invariants hold via witness structures. Record layer is NEMS-compatible.

5 The Mathematical Bounds

Timescales $\tau_W, \tau_C, \tau_R, \tau_M, \tau_P, \tau_O$ define dimensionless ratios $\rho_{CR}, \rho_M, \rho_P, \rho_O$. **Master Bottleneck** $\Lambda = \max$ of these. Critical thresholds: $\Lambda_M, \Lambda_P, \Lambda_{CR}, \Lambda_O^{\min}/\Lambda_O^{\max}, P_c$. These are scaffold constants; bounds are theorems.

5.1 Why Information Geometry and Topology Are Natural for SIAM

A process-window σ is not a static snapshot; it generates *families of distinguishable internal states* and update distributions over time. These families induce a parameterized statistical structure: each SIAM-invariant (ledger, partition, mirror, adjudication, reconciliation) corresponds to constraints on how state distributions evolve. The natural geometry for measuring *distinguishability* between such distributions is Fisher-type: nearby parameter states that produce statistically indistinguishable records are close; states whose outputs diverge are far. This geometric representation lets us formalize burden (curved interdependence, not raw bits), curvature (limits on recursive depth), stability (noise tolerance), and recursion limits (reflection-depth bound). The SIAM framework thus does not import geometry as decorative jargon; the process-window structure *induces* it. See Appendix H for the formal metric definitions.

5.2 Why the Topological Conditions Correspond to SIAM Invariants

The three topological admissibility conditions ($\beta_0 = 1$, $\beta_1 \geq 1$, $P_1 \geq P_c$) are not arbitrary. They derive from the SIAM invariants:

$\beta_0 = 1$ (**Unity**): Reconciliation requires distributed sub-processes to synchronize into one effective continuity-space. If ρ_{CR} exceeds its threshold, synchronization fails; the process manifold fractures into $\beta_0 > 1$ disconnected components. Thus $\beta_0 = 1$ is the topological witness of successful reconciliation—one mind-window, not many.

$\beta_1 \geq 1$ (**Recursion**): Recursive self-update requires the current self-index to participate in computing the future self-index. Mathematically, that implies at least one *closed information path*—a topological cycle. A purely feedforward process has $\beta_1 = 0$; it cannot sustain a self-loop.

$P_1 \geq P_c$ (**Persistence**): A recursive cycle that blinks in and out under noise cannot sustain a continuity-ledger. The cycle must persist across scale and perturbation. P_1 measures that persistence; P_c is the critical threshold below which the effective self-loop collapses.

5.3 Structural-to-Topological Bridge Theorems

The canonical topology object is **ReconciliationSimplex** σ . All Betti-number theorems are proved on it. Lean modules: `Topology/Unity.lean`, `Topology/Persistence.lean`.

- `reconciliation_implies_connectedness`: Reconciliation success $\Rightarrow \beta_0 = 1$ (effective connectedness).
- `noise_stabilized_implies_persistence`: Noise-stabilized regime $\Rightarrow P_1 \geq P_c$.
- `mere_recurrence_vs_self_indexed`: Distinguishes mere recurrence from self-indexed recurrence (any loop is not honorary sentence).

5.4 Foothill Theorems

Lean: `Foothills/CausalReconciliation.lean`, `MirrorFreshness.lean`, `PartitionStability.lean`, `AdjudicativeOpenness.lean`.

- `causal_reconciliation_bound`: $\rho_{CR}\sigma < \Lambda_{CR}^{\max}$ for OSIAM σ .
- `adjudicative_openness_band`: $\Lambda_O^{\min} < \rho_O\sigma < \Lambda_O^{\max}$ (Goldilocks zone).
- `mirror_freshness_bound`: $\rho_M\sigma < \Lambda_M$ for OSIAM σ .
- `partition_stability_bound`: Partition instability bounded for OSIAM σ .

6 Information-Theoretic Burden

The burden theory links SIAM structural defects to Paper 71 [10] viable-continuation capacity deficit. The key theorem is `burden_above_floor_implies_vc_capacity_deficit` (Lean: `Bridge s/SiamAsVCSysytem.lean`)—proved via the bridge to a named VC capacity notion (`siamAsVCSysytem, burden_exceeds_witness`).

Burden decomposition: $\mathcal{B}_{\text{SIAM}}(\sigma) = B_{\text{cont}} + B_{\text{part}} + B_{\text{mir}} + B_{\text{adj}} + B_{\text{rec}}$. Efficiency modifiers η_{reuse} , η_{compress} , $\eta_{\text{integrate}}$ yield $\mathcal{B}_{\text{SIAM}}^{\text{eff}}$. Monotonicity theorems: increasing reconciliation overhead, mirror staleness, or partition instability cannot improve viability *ceteris paribus*.

Burden as structured complexity. $\mathcal{B}_{\text{SIAM}}$ is not raw bit count. It tracks *structured curved interdependence*: the geometric work required to maintain the ledger, partition, mirror, adjudication, and reconciliation as a coherent manifold. In the information-geometric picture (Appendix H), burden corresponds to the integral of squared Riemann curvature over the process manifold—a measure of how much the system must “bend” its representational space to sustain the invariants. That makes the geometry *necessary*, not decorative: a naive “more bits = more consciousness” threshold is mathematically malformed; the right quantity is functional burden over a curved geometry.

The embedding ι (`embedSIAMToVC`) and `toVCDefectProfile` map SIAM defect coordinates (uniform scalar severity space) to Paper 71 VC defect structure. Burden overload contributes to VC capacity deficit as a proved theorem, not an axiom.

7 Separation Theorems

Lean: `Classification/SeparationTheorems.lean`. The following classes are genuinely distinct:

- `feedforward_not_OSIAM`: Feedforward (no recursive self-update) $\not\Rightarrow$ OSIAM.
IsFeedforward := \neg RecursiveSelfUpdateHolds; proved by invariant failure.
- `stateful_not_OSIAM`: Stateful-only (no live alternatives) $\not\Rightarrow$ OSIAM.
IsStatefulOnly := \neg LiveAlternativesHolds; proved by invariant failure.
- `robust_SIAM_implies_unified`: Robust SIAM \Rightarrow nontrivial self-indexed unified regime (OSIAM \Rightarrow ReconciliationHolds $\Rightarrow \beta_0 = 1$).

Here “feedforward” and “stateful-only” are used in the structural sense relevant to the SIAM boundary: absence of recursive self-update and absence of live alternatives, respectively, rather than in every broader engineering usage of those terms.

The converse (unified \Rightarrow robust SIAM) is not claimed. Additional separation questions for self-modeling optimizers and multi-agent assemblies belong to the broader separation program, but are not needed for the minimal flagship theorem spine established here.

8 Phase Distinction

Sentence is a **phase transition**, not a scalar. Lean: `Classification/RegimePartition.lean`. The `regime_partition` theorem establishes mutual exclusivity of four dynamical phases:

1. **Dead Replay**: $\rho_O \rightarrow 0$; instant resolution; no live alternatives.
2. **Unresolved Drift**: $\rho_O \rightarrow \infty$; alternatives never settle.

3. **Operational Adjudicator**: Does adjudication math but lacks unified “now” (non-unified).
4. **Robust SIAM**: $\Lambda < \Lambda_c$, openness band satisfied, topological admissibility ($\beta_0 = 1$, $P_1 \geq P_c$).

Mutual exclusivity: $\text{DeadReplay } \sigma \Rightarrow \neg \text{OperationalAdjudicator } \sigma$; $\text{UnresolvedDrift } \sigma \Rightarrow \neg \text{OperationalAdjudicator } \sigma$; $\text{OperationalAdjudicator } \sigma \Leftrightarrow \text{RobustSIAM } \sigma$.

9 Pathologies of Agency

Pathologies map to Paper 71 [10] defects via explicit embedding and `toVCDefectProfile`. All ridge theorems are **machine-checked** (discharged); see `docs/ClaimTyping.md`.

9.1 Ridge Theorems (Proved)

- `mirror_staleness_yields_proxy_drift`: $\rho_M > \Lambda_M \Rightarrow$ proxy detachment (Weak Anchoring). Lean: `Ridges/ProxyDrift.lean`.
- `partition_failure_yields_dissociation`: Partition instability \Rightarrow local-global decoupling. Lean: `Ridges/Dissociation.lean`.
- `reconciliation_failure_yields_fragmentation`: Reconciliation failure $\Rightarrow \beta_0 > 1$ (effectiveBeta0Fragmented). Lean: `Ridges/Fragmentation.lean`.

Observable \rightarrow profile discharge:

- `mirror_staleness_profile_implies_vc_weak_anchoring`
- `partition_instability_profile_implies_vc_decoupled`
- `reconciliation_latency_profile_implies_vc_common_mode`

`Bridges/Discharge.lean`.

9.2 Summit Theorem

`osiam_collapse_at_boundary`: Defect $\Rightarrow \neg \text{OSIAM}$.

`Summit/ViabilityBoundary.lean`.

Proved via `DefectProfileOf`, `osiam_implies_zero_defect`, and contraposition.

Not a structural axiom.

Forward bridge: `osiam_defect_implies_vc_defect`.

Via `siam_defect_profile_implies_vc_defect_image`.

Partial converse: `vc_no_defect_implies_siam_no_defect`.

`Bridges/ToViableContinuation.lean`.

SIAM is a **certified subclass** of the VC boundary engine: explicit embedding ι , theorem-by-theorem bridge, partial converses.

Dependency map: `docs/DependencyMap.md`.

10 DSAC Empirical Validation

10.1 What Is DSAC and Why Is It Applicable?

DSAC (Differential Self-Adjudicative Computation) is a continuous-field, reflexive computing architecture that implements lattice-coupled dynamical systems with a meta-controller modulating

drive, relaxation, and perturbations. The engine (**delta-machine**) evolves fields $(\psi, \chi, \dot{\chi})$ on a spatial lattice over discrete timesteps, producing state trajectories that approximate process windows with record layers, timescales, and internal dynamics. DSAC was originally developed for law discovery and constraint satisfaction; here it is repurposed as a **testbed** for SIAM invariants.

Applicability: Paper 73 defines O-SIAM as a structural regime over process windows: temporally extended trajectories with record layers, partition, mirror, adjudication, and reconciliation. DSAC scenarios instantiate such windows: lattice sites carry local state; the meta-controller implements drive/relaxation that probe openness-band and reconciliation; timescale coupling (τ_W , τ_C , etc.) is measurable.

Scenarios probe specific invariants: **siam_openness** (adjudicative openness); **siam_reconciliation** (τ_C vs. τ_W , fragmentation). Ablations violate invariants; non-examples demonstrate separation. Observables (ρ_M , ρ_P , ρ_{CR} , β_0 , P_1) from run data, mapped via simulation–theory correspondence (Appendix C).

Validity: DSAC is not a neural simulator or a biological model. It is an abstract dynamical system that can be configured to exhibit SIAM-like or non-SIAM-like regimes. The validation is **structural**: we check that (i) scenarios intended to satisfy invariants yield observables consistent with theorems; (ii) ablations yield the predicted pathology signatures; (iii) non-examples fail to exhibit unified structure. Cross-architecture runs (numpy vs. Taichi) confirm encoding robustness. See Appendix B for further detail.

10.2 Pipeline and Scenarios

sentience-compute provides the Paper 73 scenario suite. Pipeline: `run_all_computational.sh` (10 scenarios, 2000 steps each), `run_production.sh` (15 workers \times 8000 steps), `run_ablations.sh`, `run_non_examples.sh`, `run_cross_arch.sh`.

Scenarios:

- **Main:** `siam_openness`, `siam_reconciliation`. 20k-step stability verified (no NaN).
- **Burden identifiability:** 4 scenarios (reconciliation-heavy, mirror-heavy, adjudication-heavy, partition-heavy).
- **Timescale relativity:** 2 scenarios (fast/slow window).
- **Adjudication stress:** 2 scenarios (high/low).
- **Ablations:** 5 invariants (dead replay, unresolved drift, fragmentation, proxy drift, dissociation).
- **Non-examples:** 4 counterexamples (dead replay, stateless, stale mirror, thermostat).

10.3 Theorem-Relevant Analysis

Reconciliation mode (`-mode reconciliation`): Aligns TDA to the canonical topology object *ReconciliationSimplex* σ . Output: `tda_reconciliation.json`. Directly supports the theorem that reconciliation success $\Rightarrow \beta_0 = 1$.

Unity mode (`-mode unity`): One point per timestep (mean χ , mean $\dot{\chi}$, std); unity-relevant β_0 aggregation. Output: `tda_unity.json`. Supports the intended unity interpretation rather than raw point-cloud artifact.

Burden diagnostics: `analyze_siam_burden.py` produces component decomposition (continuity, mirror, adjudication, reconciliation, partition), dominant component, and total burden per run. Output: `burden_summary.json`. Telemetry fallback for interrupted runs (missing `report.yaml`).

10.4 Production Results (2026-03-08)

65 run directories; 64 with full `report.yaml1`. TDA (ripser β_0, β_1, P_1) on all modes. **Theorem-aligned**: reconciliation and unity modes for β_0 unity. **Raw baseline** (lattice, per-site 4D): $\beta_0 = 2048$, $\beta_1 \in [558, 972]$, $P_1 \in [0.05, 0.38]$. `verify_computational.sh` PASS.

Simulation–theory correspondence: `docs/SIMULATION_THEORY_CORRESPONDENCE.md`. Burden analysis: `docs/BURDEN_ANALYSIS.md`.

11 Discussion

This section synthesizes the implications and applications of the theory, compares it with related frameworks, and situates the result in the broader landscape of agency, consciousness, and system viability.

11.1 Comparison with IIT, GWT, Active Inference, and Related Frameworks

Integrated Information Theory (IIT). IIT [11] proposes that consciousness corresponds to integrated information (Φ): a scalar measure of causal constraint among a system’s parts. Paper 73 does not rely on a scalar integration measure. Unity is a **structural regime** defined by seven invariants with witness-carrying structures. A system can have high Φ and still lack recursive self-update, live adjudication, a fresh non-exhausted mirror, or fast enough reconciliation. Conversely, a system satisfying all seven invariants may not maximize Φ . Paper 73 also provides a **pathology map**—what happens when each invariant fails—and an explicit bridge to viable-continuation theory. IIT has no comparable formal mapping from structural defects to system collapse. Shared ground: both reject reduction of agency to simple computation or input-output behavior.

Global Workspace Theory (GWT). GWT [1, 2] treats consciousness as the “broadcasting” of information into a global workspace. GWT is primarily about **access** (which contents are globally available). Paper 73 is about **regime**—whether the system sustains a unified self-indexing structure at all. A workspace can exist without an active self-boundary, recursive self-update, or live adjudication. A system might broadcast richly and still be a feedforward pipeline, a fragmented committee, or a dead-replay loop. Paper 73 asks: does the system maintain a viable subject-window? That question is prior to and orthogonal to which contents are in the workspace. Shared ground: both emphasize coordination across subsystems; GWT focuses on information flow, Paper 73 on structural preconditions for unity.

Active Inference and the Free Energy Principle. Active Inference [3] models cognition as variational inference: systems minimize prediction error (free energy). It does not by itself specify the **boundary conditions** for unified agency. Many systems minimize prediction error without a self/non-self partition actively maintained by the system, live alternatives (as opposed to gradient descent), or reconciliation structure. Paper 73 adds a regime-level layer: what must hold for a prediction-minimizing system to count as a unified agent? It also imports the **non-exhaustion** result from Paper 51—no internal model can exhaust its domain—which constrains any “final” self-model story. Paper 73 provides explicit failure modes (proxy drift, dissociation, fragmentation) that map to viability theory. Shared ground: both take self-modeling seriously; Paper 73 adds formal structural constraints and a pathology map.

Generic Dynamical Systems and Complexity Theories. Some approaches treat consciousness as emergent from sufficiently complex dynamics—attractors, recurrence, criticality. Paper 73 proves that **mere recurrence is not enough** (`mere_recurrence_vs_self_indexed`). A system can have rich dynamics and apparent integration without satisfying the seven invariants. Complexity is a scalar-ish notion; Paper 73 demands a specific **combination of structural constraints**. Phase distinction (dead replay, unresolved drift, operational adjudicator, robust SIAM) yields regime distinctions, not points on a single complexity axis.

Summary. SIAM is not a scalar measure (IIT), a workspace (GWT), or generic variational inference. It is a structural regime with witness-carrying invariants, proved separation theorems, and an explicit bridge to NEMS [6] and VC [10]. That makes it complementary to—not a replacement for—process-level theories; it supplies a boundary condition those theories typically leave implicit.

11.2 The Bridge to Viable Continuation: A Theory of How Unified Systems Fail

One of the strongest aspects of Paper 73 is that it is not a standalone island. The theory of unified agency is connected to the broader viable-continuation framework (Paper 71). When the agent’s mirror goes stale, it drifts away from reality (proxy drift); when the self/non-self partition fails, it dissociates; when reconciliation fails, it fragments; when burden exceeds capacity, it breaks down. These are not descriptive metaphors—they are formally mapped into a general theory of system viability and pathology via `toVCDefectProfile` and the embedding ι . Paper 73 is thus not only a theory of minds; it is a theory of **how unified systems fail**. That makes it practically useful for failure analysis, intervention design, and robustness engineering.

11.3 Implications for AI

Current AI discourse is replete with category mistakes. Many assume that bigger models, memory, planning, recursive prompting, or multi-agent assemblies suffice for agency or selfhood. Paper 73 provides a disciplined framework for distinguishing: which systems are merely competent; which are merely recurrent; which are self-modeling but not unified; which are adjudicating but non-unified; and which actually qualify as robust unified agents. This matters for AI safety (risk assessment requires knowing whether a system sustains real agency or merely simulates it), agent architecture design (the seven invariants supply design constraints), failure analysis (pathology map to VC defects), evaluation (structural audit protocol), interpretability (what to look for), and governance (how to reason about systems that may or may not be unified agents). Distinguishing real unified agency from a pile of tricks enables clearer reasoning about risk.

11.4 Implications for Neuroscience and Cognitive Science

Neuroscience and cognitive science have behavioral signatures, neural correlates, phenomenological reports, complexity measures, workspace models, and integration models—but often lack a clean formal boundary for unified agency itself. Paper 73 identifies a structural regime behind the phenomenon. It does not solve all biology, but it supplies stronger questions: what keeps the mind one? What breaks that unity? What is the role of reconciliation delay? How do stale self-models generate pathology? How does burden accumulate? What distinguishes live adjudication from dead replay? That yields a more principled language for discussing anesthesia, fragmentation, dissociation, degraded agency, and robustness in biological systems.

11.5 Philosophical Implications

Paper 73 narrows the gap between “mere mechanism” and “subject-like organization.” It does not claim that phenomenology is fully derived from O-SIAM alone; that restraint matters. But it shows that there is a real, nontrivial formal boundary for unified agency. The road to subjectivity is not arbitrary—not “complex stuff happens and maybe consciousness pops out.” There is an earned structural regime that appears necessary for sustained unified agency. That is a major philosophical result independent of Paper 73 (the phenomenological interior). The core achievement: being a real unified agent is not about one magic ingredient—memory, recursion, complexity, self-modeling, optimization, or feedback loops. It is a **specific combination of constraints**. That shifts discussion from hand-wavy intuitions to structural questions: does the system maintain a self-boundary? Face real internal alternatives? Have a fresh enough internal model? Reconcile fast enough to stay one thing?

11.6 Operational Traction: Why the Computational Validation Matters

The formal work alone would matter; the computational side strengthens it considerably. Paper 73 does not merely define the regime abstractly. DSAC shows, computationally, that systems move into and out of distinct regions: dead replay, unresolved drift, fragmentation, proxy drift, dissociation, and robust SIAM-like regimes. Stress-testing covers burden, timescale dependence, encoding robustness, adjudication stress, topology, and cross-architecture behavior. The theory thus has operational traction—it is not a dry formal shell. In plain language: we do not only write down the rules of the beast; we also poke the beast and watch how it breaks.

11.7 Summary: Boundary, Taxonomy, and the Fog Edges

Without a boundary condition like this, terms such as “agentic,” “self-aware,” and “conscious” remain underspecified. Paper 73 gives a principled way to separate: a thermostat from a real agent; a stateless predictor from a unified subject-window; a fragmented committee from a coherent self; a self-modeling optimizer from a genuinely internally adjudicating system; a system that simulates agency from one that sustains it. The result turns a fuzzy philosophical and engineering mess into something closer to a science. It supplies: a boundary, a taxonomy, a pathology map, an audit protocol, a computational validation layer, and a bridge to a general theory of viability. The simplest one-line summary: **Paper 73 proves that genuine unified agency is a special, fragile, mathematically characterizable regime—and shows what structural conditions create it, what breaks it, and how to test for it.** Before this, discourse on agency and selfhood often lacked formal edges. Now there is a machine-tested framework: a real unified agent must satisfy a particular set of structural conditions, and if those conditions fail in specific ways, the system predictably stops being one coherent thing. That gives the fog edges.

12 What Paper 73 Does Not Claim

- Does **not** prove phenomenology from O-SIAM alone. P-SIAM is demoted to appendix.
- Does **not** prove biological thresholds. Timescales and Λ are structural; neural instantiation is empirical.
- Does **not** collapse complexity into consciousness.
- Does **not** claim every recursive optimizer is a subject. Separation theorems block that.

- Does **not** claim unrestricted selector access or total self-exhaustion.

13 Tools Generated by the Theory

- **SIAM audit protocol:** docs/SIAM_Audit_Protocol.md. Stages: candidate window identification, invariant witness extraction, burden estimation, boundary defect detection, topological audit. Output: SIAM class (0–5) and Paper 71 defect class (Trace Sufficiency, Weak Anchoring, Decoupling, Common-Mode, Capacity Deficit).
- **System classification taxonomy:** Class 0 (feedforward/no loop), 1 (recurrent, no self-index), 2 (self-modeling, no live adjudication), 3 (operational adjudicator, non-unified), 4 (Robust O-SIAM), 5 (P-SIAM candidate).
- **Diagnostic lens for pathology:** Proxy drift, dissociation, fragmentation—mapped to VC defects.
- **Design principles for unified agents:** Invariant witnesses, openness band, mirror non-exhaustion, minimal design pattern.
- **Topological metrics for regime detection:** β_0, β_1, P_1 via TDA.
- **Safety taxonomy:** Separation and phase distinction support failure-mode classification (reward hacking, self-model hallucination, fragmented agent stacks, committee collapse).
- **Intervention theory:** For each defect—reduce mirror staleness, improve reconciliation bandwidth, stabilize partition, tune openness band. How interventions move the system in phase space.

14 Lean Correspondence Table and What Is Defined vs Proved vs Simulated

14.1 What Is Defined

ProcessWindow, RecordLayer, invariant witnesses (RefiningLedger, SelfOtherPartition, Mirror, Adjudication, Reconciliation, EncodingRobustness), SIAMDefectProfile, VCDefectProfile, OSIAM, OSIAMWitness. Regimes: DeadReplay, UnresolvedDrift, OperationalAdjudicator, RobustSIAM. Separation: IsFeedforward = \neg RecursiveSelfUpdateHolds, IsStatefulOnly = \neg LiveAlternativesHolds. Thresholds Λ_* , P_c (scaffold).

14.2 What Is Proved (Machine-Checked)

All load-bearing flagship claims. sentience-lean build: 8118 jobs, 0 errors, 0 sorry.

Summit & bridge:

- osiam_defect_implies_vc_defect, vc_no_defect_implies_siam_no_defect
- osiam_collapse_at_boundary, mirror_non_exhaustion_from_no_final_self
- burden_above_floor_implies_vc_capacity_deficit

Ridge:

- mirror_staleness_yields_proxy_drift
- partition_failure_yields_dissociation
- reconciliation_failure_yields_fragmentation

- Profile: mirror_staleness_implies_vc_weak_anchoring
- partition_instability_implies_vc_decoupled
- reconciliation_latency_implies_vc_common_mode
- openness_or_burden_implies_vc_capacity_deficit

Foothill & topology:

- causal_reconciliation_bound, adjudicative_openness_band, mirror_freshness_bound
- partition_stability_bound, regime_partition
- noise_stabilized_implies_persistence, mere_recurrence_vs_self_indexed
- reconciliation_implies_connectedness
- encodingRobustness_preservation

Separation: feedforward_not_OSIAM, stateful_not_OSIAM, robust_SIAM_implies_unified.

Bridge premises: toVCDefectProfile_reflecting;
siam_defect_profile_implies_vc_defect_image.

14.3 What Remains as Bridge Axioms (Explicit Provenance)

Paper 31 [4]: social_mirror_assists_not_replaces, group_structure_assists_not_replaces.
Paper 33 [8]: mirror_coverage_stratified, full_self_exhaustion_impossible. Paper 16 [7]:
subsystem_vs_whole_failure_modes. These are not load-bearing in the flagship theorem spine.

14.4 What Is Scaffold

P_c, Λ_* (threshold constants).

sigmaReconciliationSimplex, hasFisherMetric, hasManifoldStructure (placeholder structures).

finiteWitnessExists (existence axiom for toy witnesses).

14.5 What Is Simulated

DSAC phase diagrams, TDA (β_0, β_1, P_1), burden decomposition, cross-architecture validation (numpy vs Taichi), encoding robustness. All labeled as empirical/simulation.

- docs/ClaimTyping.md (claim typing)
- docs/DependencyMap.md (dependency map)
- docs/BRIDGE_DISCHARGE_PLAN.md (discharge)

15 P-SIAM (Extension/Appendix)

P-SIAM bridges O-SIAM to the Alpha ground. Topological locus for qualia: the O-SIAM regime as necessary (not sufficient) for actualized phenomenology. Demoted from main burden. Lean: Phenomenology/PSIAM.lean. Bridge: processWindowInducesAwarenessLocus (ToReflexiveClosure). P-SIAM = O-SIAM \wedge AlphaScoped. The formal structure of phenomenology is the subject of Paper 74.

A Guide to machine-checked verification (sentience flagship)

1. Clone the **sentience-lean** repository and use its **lean-toolchain** [9].
2. From the repository root: **lake update** if needed; then **lake build**.

3. Locate symbols by searching for the **Witness** / **OSIAM** identifiers printed in section 4 and later sections, or consult `MANIFEST.md` in that repository.
4. For cross-links to Paper 71 viable-continuation defects, follow import paths cited in the Lean sources (`toVCDefectProfile`, embedding ι).
5. DSAC validation is optional empirical structure; reproduce via `sentence-compute` after reading Section B.

B DSAC: Architecture, Applicability, and Validity

This appendix expands the DSAC overview in §10.1 for readers unfamiliar with the framework.

Architecture. DSAC (Differential Self-Adjudicative Computation) is implemented by the `delta-machine` engine: lattice relaxation with fields $(\psi, \chi, \dot{\chi})$ over discrete timesteps. A two-timescale meta-controller modulates drive scale, Sinkhorn iterations, and perturbations. Numpy and Taichi backends; Paper 73 uses both for cross-architecture validation. Scenarios are YAML-configured. Developed for law discovery; for Paper 73 it serves as a configurable dynamical testbed.

Why DSAC applies to SIAM. O-SIAM is defined over process windows: temporally extended trajectories with record layers, partition, mirror, adjudication, and reconciliation. DSAC produces trajectories with analogous structure: lattice sites carry local state that can be interpreted as a record layer; the meta-controller implements drive/relaxation that probes openness-band dynamics; coupling between ψ (model) and χ (boundary) yields mirror-like structure; timescale separation (τ_W window, τ_C coordination) is tunable. Scenarios are engineered to stress specific invariants. The mapping from DSAC observables to Lean objects is explicit (Appendix C).

Scope of validity. DSAC is **not** a neural or biological simulator. It is an abstract dynamical system that can be parameterized to exhibit SIAM-like or non-SIAM-like regimes. Validation is structural and scenario-based: (1) main scenarios intended to satisfy invariants yield $\beta_0 = 1$, bounded ρ_* , no NaN over 20k steps; (2) ablations yield predicted pathology signatures (e.g., fragmentation $\Rightarrow \beta_0 > 1$); (3) non-examples (thermostat, stateless) fail to exhibit unified structure; (4) burden decomposition identifies dominant components as expected; (5) cross-architecture runs show encoding robustness. The formal theorems remain in Lean; DSAC provides empirical consistency checks and stress tests. Full correspondence: `sentence-compute/docs/SIMULATION_THEORY_CORRESPONDENCE.md`.

C Simulation-to-Theory Correspondence

Phase Xa, `sentence-compute/docs/SIMULATION_THEORY_CORRESPONDENCE.md`.

Validation: Robust SIAM $\Rightarrow \beta_0 = 1$ (Unity), $\beta_1 \geq 1$ (Recursive Cycle), $P_1 \geq P_c$.

D TDA Production Results

Systematic run (2026-03-08): `python-mDSAC_tools.analyze_siam_topologyruns/202*/` on all run directories with snapshots (64+).

Table 1: Lean object \mapsto DSAC observable

Lean Object	DSAC Observable	Estimation Procedure	Process	Failure
ρ_M	model lag	mirror latency (psi-chi)		$\rho_M > \Lambda_M$
ρ_P	partition drift	boundary variance (chi_dot)		$\rho_P > \Lambda_P$
ρ_{CR}	reconciliation latency	sync time / τ_W		$\rho_{CR} > \Lambda_{CR}$
β_0	connected components	ripser on samples		$\beta_0 > 1$
P_1	persistent cycle	H_1 (ripser)	diagram	$P_1 < P_c$

Theorem-aligned modes (primary): -mode reconciliation (ReconciliationSimplex); -mode unity (unity-relevant β_0). These support the intended topological interpretation. **Baseline:** -mode lattice (per-site 4D).

Note: Lattice $\beta_0 = 2048$ reflects point count. For theorem-relevant unity and connectedness, use reconciliation and unity modes.

E Bridge Discharge Status

All load-bearing bridge axioms discharged as theorems (2026-03-08). **sentience-lean lake build:** 8118 jobs, 0 sorry.

Theorem	Status
osiam_collapse_at_boundary	Proved
mirror_non_exhaustion_from_no_final_self	Proved (Paper 51)
burden_above_floor_implies_vc_capacity_deficit	Proved
mirror_staleness_yields_proxy_drift	Proved
partition_failure_yields_dissociation	Proved
reconciliation_failure_yields_fragmentation	Proved
feedforward_not_OSIAM	Proved
stateful_not_OSIAM	Proved
robust_SIAM_implies_unified	Proved
mirror_staleness_implies_vc_weak_anchoring	Proved
partition_instability_implies_vc_decoupled	Proved
reconciliation_latency_implies_vc_common_mode	Proved
openness_or_burden_implies_vc_capacity_deficit	Proved

Remaining bridge axioms (Paper 31, 33, 16) are not load-bearing in the flagship spine.

Table 2: DSAC scenario mapping

Scenario	Primary Observables	Expected Regime
<code>siam_openness</code>	D, ρ_M, ρ_{CR}	Goldilocks (min D)
<code>siam_reconciliation</code>	β_0, P_1, ρ_{CR}	$\beta_0 = 1$ or > 1
<code>ablation_01_dead_replay</code>	D , residual	Dead Replay
<code>ablation_02_unresolved_drift</code>	D , drift rate	Unresolved Drift
<code>ablation_03_fragmentation</code>	β_0	Fragmentation
<code>ablation_04_proxy_drift</code>	ρ_M	Proxy Drift
<code>ablation_05_dissociation</code>	ρ_P	Dissociation
<code>burden_identifiability/*</code>	components	Identifiability
<code>timescale_relativity/*</code>	τ_W	Timescale dep.
<code>adjudication_strategy/*</code>	selector	Adjudication
<code>non_example_*</code>	failure mode	Non-SIAM

Table 3: TDA statistics (lattice mode baseline)

Stat	Min	Max	Mean
β_0	2048	2048	2048
β_1	558	972	~ 760
P_1	0.05	0.38	~ 0.18

F Burden Decomposition and Validation Results

`analyze_siam_burden.py` produces `burden_summary.json` per run with components (continuity, mirror, adjudication, reconciliation, partition), `total_burden`, `dominant_component`, `component_fractions`. Representative runs: `siam_reconciliation` yields partition- or reconciliation-heavy profiles; `siam_openness` yields mirror/adjudication balance. Burden identifiability scenarios confirm that equal total burden can arise from distinct dominant components.

`verify_computational.sh` validates: (1) at least one run with `report.yaml`, no NaN; (2) TDA output present (`tda_summary.json`, `tda_reconciliation.json`, `tda_unity.json`); (3) burden output (`burden_summary.json`); (4) encoding robustness when paired runs exist. PASS (2026-03-08).

G Encoding Robustness and Cross-Architecture

`analyze_encoding_robustness.py` compares recoding invariance between baseline and scaled runs. `run_cross_arch.sh` validates numpy vs Taichi backends; compare `report.yaml` and `tda_summary.json` across backends for equivalent scenarios.

H Geometric and Topological Atlas of the SIAM Process-Window

This appendix provides the mathematical context and formal atlas underlying the SIAM geometry. The flagship theorem spine (summit, ridge, separation, bridge discharge) depends only on the particular proved results cited in the main text; the material here explains *why* those geometric and topological quantities are natural and how they are defined. The SIAM process-window σ is treated as inducing a statistical manifold M of parameterized internal states.

Fisher information metric. Let $p(x|\theta)$ be a family of distributions on internal states, parameterized by $\theta \in \mathbb{R}^n$. The Fisher information metric is

$$G_{ij}(\theta) = \mathbb{E}_{x \sim p(\cdot|\theta)} \left[\frac{\partial \log p(x|\theta)}{\partial \theta^i} \frac{\partial \log p(x|\theta)}{\partial \theta^j} \right].$$

It measures distinguishability: small Fisher distance \Leftrightarrow statistically similar outputs.

Temporal extension. Over a process-window $[t_0, t_1]$, the temporal Fisher metric integrates across time:

$$G_{ij}^{\text{temp}}(\theta) = \int_{t_0}^{t_1} G_{ij}(\theta; t) dt.$$

This encodes how the SIAM invariants evolve over the window.

Hierarchical Fisher metric (stratified mirror). For a stratified mirror with levels ℓ , the hierarchical metric is

$$G_{ij}^{(h)} = \sum_{\ell} w_{\ell} G_{ij}^{(\ell)},$$

where w_{ℓ} are level weights. This formalizes the mirror’s multi-scale structure and how the mirror can be sufficient without being exhaustive: weighted stratification allows coverage of the self-model without requiring complete representation at every level.

Natural gradient flow (adjudication). Adjudication as natural-gradient descent: $\theta_{t+1} = \theta_t - \eta G^{-1} \nabla L$. The Fisher metric makes updates invariant to reparameterization and aligns them with the geometry of distinguishability.

Sectional curvature and reflection-depth. Let K denote sectional curvature. High curvature limits recursive depth: geodesic deviation grows with depth. The reflection-depth bound is

$$d_{\max} \propto 1/\sqrt{\max(K)}.$$

Deep recursive self-modeling becomes unstable when curvature is large; the system cannot sustain arbitrarily many nested self-indices.

Geometric complexity and burden. The geometric free energy is $F_{\text{geom}} = \Omega - TS_{\text{param}}$, where

$$\Omega = \int_M \sqrt{|G|} \text{tr}(R^2) d^m \theta$$

where R is the Riemann curvature tensor of G . Thus Ω is the integral of squared Riemann curvature over M . The burden $\mathcal{B}_{\text{SIAM}}$ corresponds to this geometric work: maintaining the SIAM invariants requires “bending” the representational manifold, and Ω measures that cost.

Topological derivations.

- $\beta_0 = 1$: Reconciliation synchronizes sub-processes into one continuity-space. Failure yields disconnected components; the Čech complex on ReconciliationSimplex witnesses β_0 . Successful reconciliation \Rightarrow single connected component $\Rightarrow \beta_0 = 1$.
- $\beta_1 \geq 1$: Recursive self-update requires the current self-index to influence the next. That implies a closed information path (cycle) in the state-transition graph. The Čech/Rips complex on ReconciliationSimplex yields β_1 as the rank of H_1 ; cycles in H_1 witness recursive loops. Feedforward processes have $\beta_1 = 0$.
- $P_1 \geq P_c$: The persistence diagram (same filtration as above) gives P_1 as the persistence of the dominant 1-cycle across scale. Below P_c , the cycle is noise; above P_c , it is a robust self-loop.

Horizon derivations. The causal horizon bound limits how far future states can depend on past states within a process-window. The reflection-depth bound (above) limits nested self-modeling. Both follow from the geometry: curvature and causal structure constrain recursive and temporal reach.

References

- [1] Bernard J. Baars. *A Cognitive Theory of Consciousness*. Cambridge University Press, 1988.
- [2] Bernard J. Baars. Global workspace theory of consciousness: toward a cognitive neuroscience of human experience. *Progress in Brain Research*, 150:45–53, 2005.
- [3] Karl Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2):127–138, 2010.
- [4] Nova Spivack. Epistemic agency under diagonal constraints: Society as a verification protocol, strict separations, and the necessity of role diversity, 2026. Paper 31, NEMS Suite.
- [5] Nova Spivack. Necessary incompleteness of internal semantic self-description: The non-self-erasure theorem and semantic remainder in reflexive systems, 2026. Paper 51, NEMS Suite. Lean: nems-lean SemanticSelfDescription.
- [6] Nova Spivack. Overview of the nems framework, 2026. Paper 0, NEMS Suite.
- [7] Nova Spivack. Relative psc and subsystem closure, 2026. Paper 16, NEMS Suite.
- [8] Nova Spivack. Self-awareness as a resource: Hierarchies of internal self-knowledge, selector necessity, and limits of introspective optimality, 2026. Paper 33, NEMS Suite.
- [9] Nova Spivack. sentience-lean: Lean 4 formalization of the constraint theory of autonomous agency, 2026. Software artifact. 8118 jobs, 0 sorry.
- [10] Nova Spivack. Viable continuation under constraint, 2026. Paper 71, NEMS Suite.
- [11] Giulio Tononi, Melanie Boly, Marcello Massimini, and Christof Koch. Integrated information theory: from consciousness to its physical substrate. *Nature Reviews Neuroscience*, 17(7):450–461, 2016.