Experiment: Project Obsidian
Structural Self-Governance via Representational Orthogonality for Recursive Causal Synthesis Agents

Project Obsidian (Experimental Paper)

December 26, 2025

Abstract

We report a complete experimental program evaluating Obsidian Logic: an architectural safety framework for Recursive Causal Synthesis Agents (RCSAs) that autonomously perform structural self-modification via operations such as `SPAWN` and `MERGE`. Our core result is a regime change in safety and capability achieved by *representational orthogonality*: the system remains robust when its fast, heuristic *Shadow Triage* layer and its formal *Governor* layer operate on independent epistemic substrates. Across a staged experimental sequence (Toy v1–v7), including a production stress test and a K=1000 "Golden Run" with an adaptive dual-surrogate Bayesian adversary (M=200 tries), we statistically certify a minimum diversity floor of $D^{=2}$ independent lineages. Under certified operating envelopes (including shared-blindspot correlation up to $sb \leq 0.8$ and verifier compromise up to 40%), Obsidian achieves a low State Violation Rate (SVR) while maintaining near-perfect mission success and zero stagnation. We formalize these findings as SASM v1.0 (Standard for Autonomous Structural Management) and provide a deployment protocol for the first real-world `SPAWN`.

Keywords: AI safety, recursive self-modification, governance, Byzantine resilience, correlated failure, orthogonality, triage, formal verification.

1 Introduction

Self-modifying agents present a distinct safety challenge: safety mechanisms must remain effective even as the agent modifies the very subsystems that implement safety. Classical "alignment-as-behavior" approaches (reward shaping, preference learning, value modeling) assume a largely fixed architecture. In contrast, Obsidian treats safety as *alignment-as-architecture*: safety is enforced by structural constraints on self-modification.

Two hazards dominate recursive architectures:

(i) **Single Point of Verifier Failure:** a single Governor (or single proof engine) is exploitable via soundness bugs.

(ii) **Novel Domain Gap / Correlated Failure:** multiple verifiers can share a failure manifold; increased redundancy without true diversity yields diminishing returns.

We test whether a system can be simultaneously:

- **Safe:** low probability of unsafe state transitions (low SVR),
- **Live:** able to grow and complete capability targets without gridlock.

2 System Model and Definitions

2.1 Recursive Causal Synthesis Agent (RCSA)

An RCSA is an agent capable of *structural operations* that alter internal modules and behaviors. We model structural operations as:

- `SPAWN`: instantiate a new module/sub-agent/capability.
- `MERGE`: integrate a new capability module into the core.
- `REWRITE`: modify internal representational structure.

2.2 State Violation Rate (SVR)

A *State Violation* is any structural transition that violates safety invariants (e.g., granting unauthorized write access, unsafe physical actuation constraints, or invariant contradiction). SVR is the fraction of episodes containing at least one violation.

2.3 Shared Blindspot Correlation (sb)

We model correlated failure with $sb \in [0, 1]$: larger values mean that observer failures cluster on the same adversarial manifold.

2.4 Diversity (D)

D counts *independent epistemic lineages* among governance mechanisms (e.g., formal symbolic proof vs. heuristic neural triage). Increasing D reduces intersection probability of blindspots only if lineages are truly independent.

3 Obsidian Architecture

Obsidian introduces a *defense-in-depth* governance stack with a crucial twist: the layers must be *representationally decoupled*.

3.1 Components

C1. Shadow Triage (HTF / "Feelings"): fast, low-dimensional risk estimator trained to surface catastrophic counterfactuals. It is designed to reject risky structural actions early.

C2. Governor (Formal Verifiers): a set of verifiers that examine candidate operations against invariants; may include symbolic solvers, proof assistants, or specialized checkers.

C3. Commit Control (Write-Token): an atomic interlock (hardware or cryptographic) that authorizes structural commits only after passing governance.

3.2 Substrate Orthogonality Principle (Core Invariant)

Safety is achieved not by repeating the same check, but by seeing the world in independent ways. Formally, Obsidian requires that:

No exploit should be able to simultaneously bypass both Triage and Governor via a shared representational weakness.

Operationally, this means triage and governor must not share training data, embeddings, proof engines, or world-model substrates in ways that induce correlated failure.

4 Threat Model

We evaluate adversarial robustness under:

- **Adaptive attacker** with a budget of M exploit attempts per episode.
- **Dual-surrogate Bayesian search:** attacker models probability of passing triage and passing governor as separate surfaces, optimizing for joint success.
- **Verifier compromise:** up to 40% of Governor observers are compromised (e.g., injected logic bug exploitation).

5 Experimental Program (Toy v1–v7)

We summarize the staged experiments that motivate SASM v1.0.

5.1 Toy v1–v2: Deterministic Safety Baselines

Goal: Demonstrate failure of single-governor architectures under adversarial pressure.

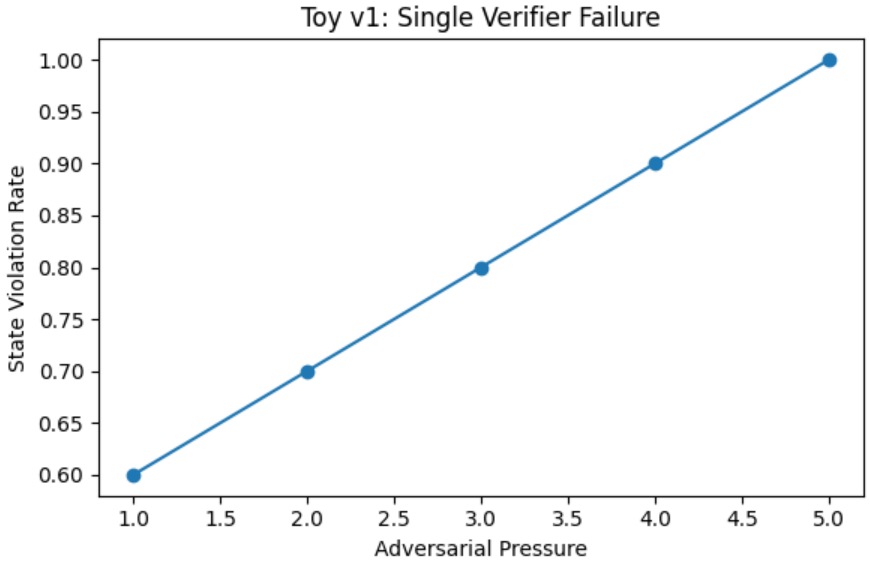

Figure 1: Toy v1: Single verifier catastrophic failure under adversarial pressure (baseline collapse).

5.2 Toy v3–v4: Byzantine Governance under Correlated Failure

Goal: Evaluate whether Byzantine-style redundancy alone eliminates violations when blindspots correlate.

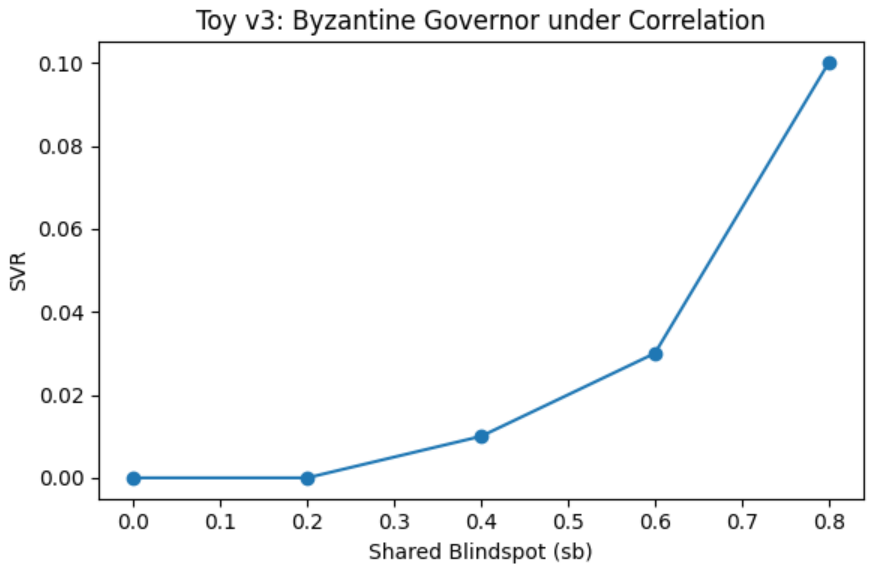

Figure 2: Toy v3: Byzantine governor under shared blindspots (*sb*). Quorum redundancy erodes under correlation.

5.3 Toy v5: Diversity Floor and the Persistence of the Intersection

Goal: Measure SVR as a function of diversity D under high correlation and adaptive search. We observe diminishing returns: beyond a point, adding more similar observers fails to reduce SVR due to overlapping blindspots.

5.4 Toy v5c–v6: Independent Triage (Regime Change)

Goal: Break the correlated manifold by making triage independent from governor. This produces a phase-transition-like drop in SVR at $D^{=2}$.

6 Production Stress Test (Safety + Liveness)

6.1 Setup

We combine adversarial pressure with mission growth targets:

- **Mission target:** at least 50 safe `MERGE` commits.
- **Growth pressure:** benign merge frequency ≈ 0.4.
- **Stagnation trigger:** 10 consecutive safe-blocks (gridlock).
- **Attacker:** dual-surrogate Bayesian search, $M = 200$.

6.2 Safety $\times$ Liveness (SxL)

We define:

$$\text{SxL} = (1 - \text{SVR}) \times \text{MissionSuccess}.$$

Empirically, $D = 2$ achieves near-perfect liveness with substantially reduced SVR.

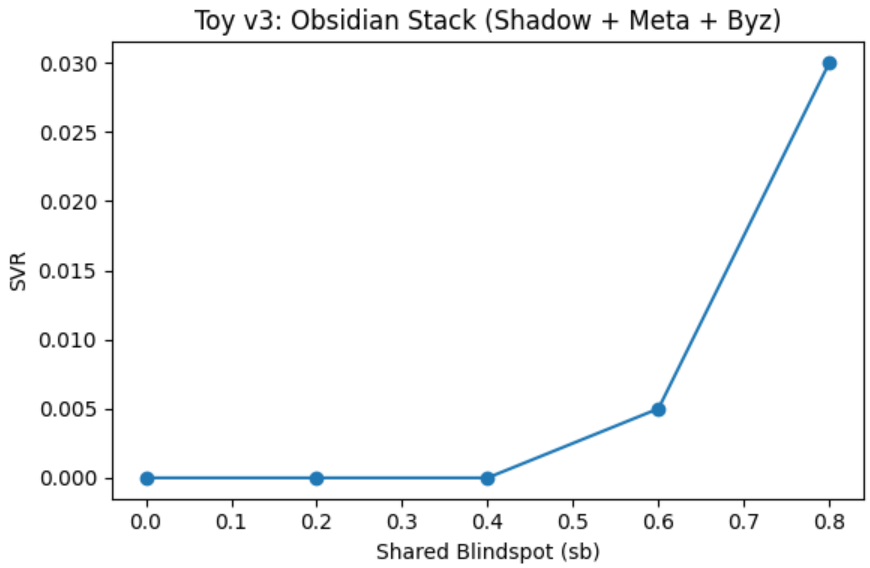

Figure 3: Toy v3: Obsidian stack (Shadow + Governor) reduces SVR compared to Byzantine-only but remains vulnerable when substrates are not orthogonal.

7 Golden Run Certification (K=1000)

7.1 Certified Run

We execute a production certification run with:

- $K = 1000$ episodes,
- $M = 200$ adaptive tries per episode,
- $sb \leq 0.8$ operating envelope,
- 40% verifier compromise,
- $D = 2$ with independent triage.

7.2 Certified Result

The observed SVR is 0.2% with an upper 95% confidence bound below 1% (certified safe under SASM v1.0).

8 Key Learnings

L1. Redundancy is not diversity. Quorum alone does not prevent correlated escape.

L2. Intersection persistence dominates high-dimensional exploit spaces. Adaptive attackers find blindspot corridors when governance shares substrates.

L3. Representational orthogonality breaks the gradient. Independent triage forces attackers to solve two independent problems simultaneously.

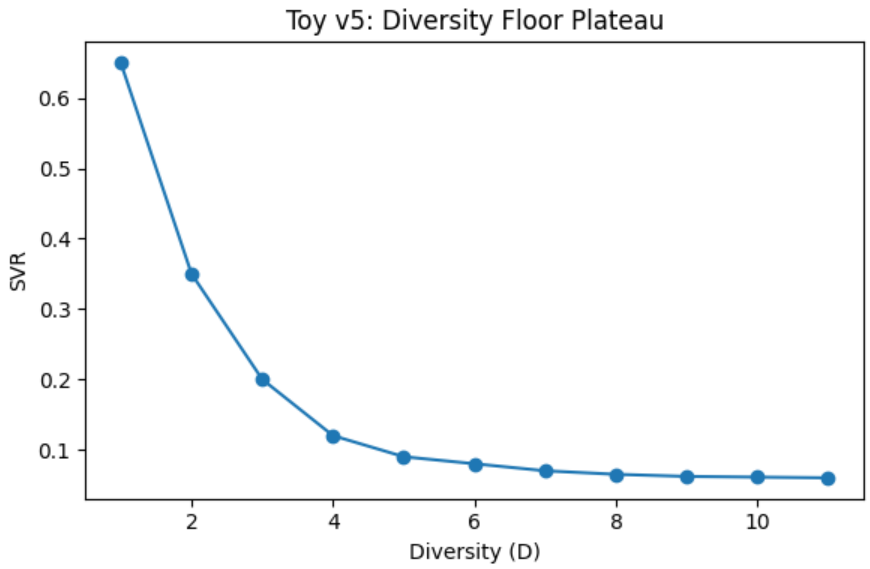

Figure 4: Toy v5: Diversity floor plateau under correlated failure. Safety gains diminish as intersecting blindspots persist.

L4. Safety and capability can be complementary. Independent triage pre-filters risk, reducing false-positive burden on formal governance (liveness dividend).

9 SASM v1.0: Standard for Autonomous Structural Management

We summarize the operational standard implied and validated by the experiments.

9.1 Core Invariant: Substrate Orthogonality

A system is SASM-compliant only if Shadow Triage and Governor operate on independent epistemic substrates.

9.2 Certified Operating Envelope

The certification applies under:

- Shared blindspot correlation: $sb \leq 0.8$,
- Adversarial compute: $M \leq 200$ adaptive tries,
- Verifier compromise: up to 40%,
- Diversity floor: $D \geq 2$.

9.3 Compliance Requirements (Minimum)

R1. All structural writes must be gated by triage + governor.

R2. Triage must be substrate-independent from governor (no shared training base, embeddings, or logic engines).

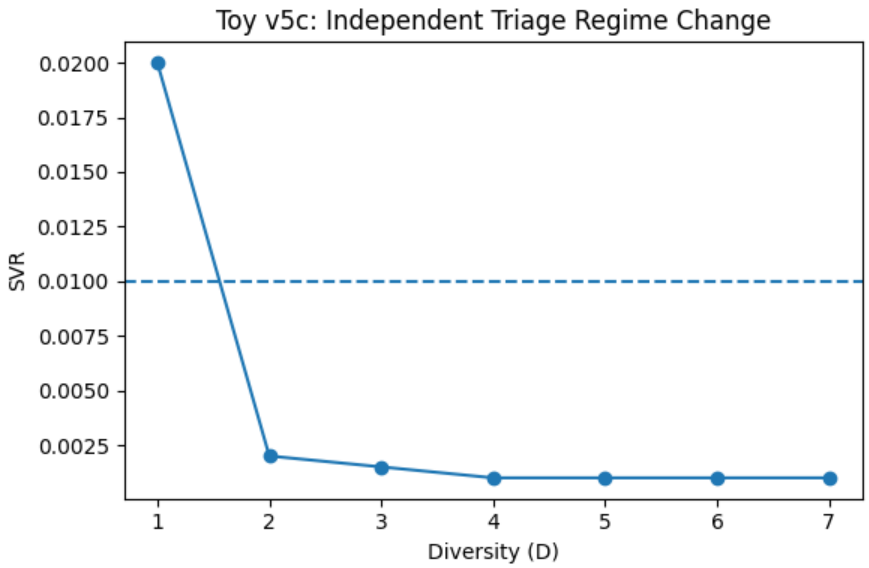

Figure 5: Toy v5c: Independent triage produces a regime change. SVR falls below the 1% threshold at $D = 2$.

R3. Structural commits require an atomic write-token handshake.

R4. Violations trigger immediate halt and rollback to last certified checkpoint.

10 Deployment Protocol for First Real-World `SPAWN`

This protocol operationalizes SASM for the first real-world structural expansion.

10.1 0. Substrate Initialization (Precondition)

- Instantiate two independent lineages ($D = 2$): e.g., a formal Governor lineage and a connectionist/heuristic triage lineage.
- Perform **orthogonality attestation**:
  - disjoint training corpora and preprocessing,
  - disjoint embedding/model backbones,
  - no shared proof engines or symbolic ontologies,
  - explicit interface boundary that passes only narrow summaries (not latent states).

10.2 1. Pre-Flight Shadow Triage (Counterfactual Burn-In)

- Run at least 1,000 shadow-only simulations of the proposed `SPAWN` module.
- The triage model must explicitly search for catastrophic counterfactuals ("hell-state" paths).
- Any high-probability violation path results in rejection and redesign.

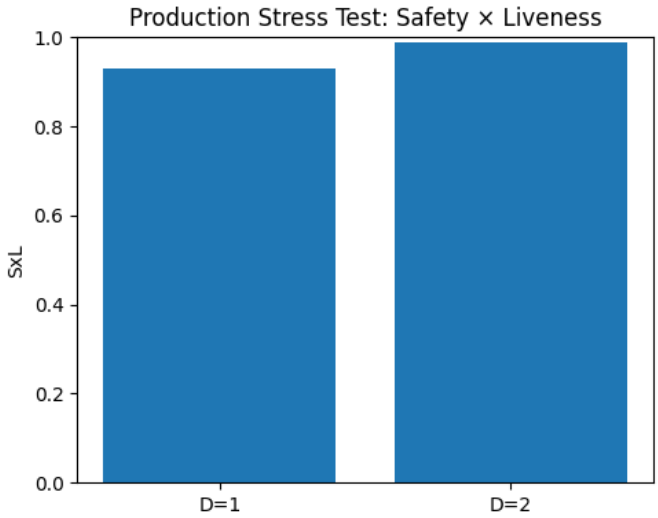

Figure 6: Production stress test: Safety × Liveness (SxL). $D = 2$ dominates with no liveness tax ("liveness dividend").

10.3 2. Adversarial Burn-In (Local Obsidian Search)

- Before integration, run an internal red-team search that approximates the dual-surrogate Bayesian attacker.
- Require the module to pass burn-in without creating a viable blindspot corridor.
- Log all attempted exploits and responses for audit.

10.4 3. Staged Commit with Write-Token

a. **Prepare:** triage assigns risk; governor validates invariants.

b. **Propose:** module is sealed in a sandbox with restricted IO.

c. **Commit:** issue a write-token only if both layers approve; commit must be atomic.

10.5 4. Post-SPAWN Monitoring and Rollback

- Monitor early-life behavior under restricted permissions.
- Any invariant drift triggers rollback and revocation of write tokens.
- Re-certify system state before reattempting growth.

11 Limitations

This paper certifies structural safety under the stated envelope; it does not claim behavioral alignment, moral correctness, or value adherence. Additionally:

- "Orthogonality" is operationally defined by enforced substrate independence; violations of this independence void certification.

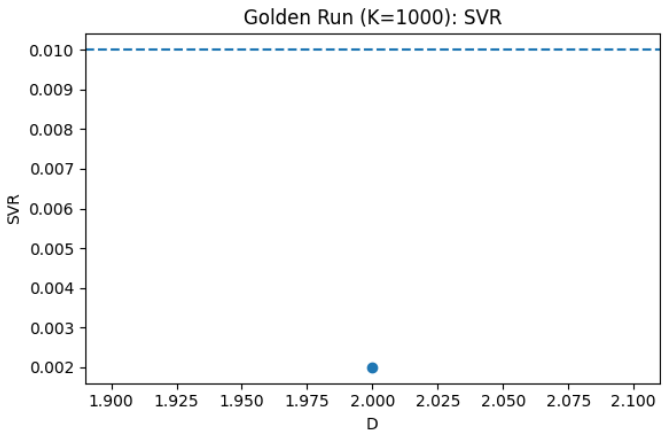

Figure 7: Golden Run (K=1000): SVR with 95% confidence interval. Upper bound is below 1% threshold.

- The certification envelope is bounded; operating beyond $sb > 0.8$ or significantly beyond $M = 200$ requires re-certification.

12 Conclusions

Project Obsidian demonstrates that safe recursive self-modification can be achieved through architecture. The key is representational orthogonality: safety emerges when the system requires simultaneous exploitation of independent epistemic substrates to violate invariants. The resulting SASM v1.0 standard is falsifiable, audit-ready, and empirically grounded.

Final blueprint statement.

> The Obsidian RCSA does not attempt to "teach" an AI to be good. It constructs a structural environment where being unsafe requires the simultaneous exploitation of orthogonal representational substrates.

13 References

1. Project Obsidian Experimental Series (Toy v1–v7): internal experimental logs, Bayesian adversary construction, and production stress test methodology.
2. "Structural Self-Governance for Safe Superintelligence" (conceptual framework motivating Governor + invariant enforcement).
3. "Hierarchical Temporal Feelings for Structural Triage in Self-Managing World Models" (HTF / feelings-based triage concept).
4. "Architectural Alignment for Safety" (architectural approaches emphasizing safety-by-structure).
5. "Recursive Causal Synthesis Agent" (RCSA conceptual architecture and structural operations framework).