Governed AI Self-Improvement

An Empirical Application of Sixfold Rot, Mechanism Discovery, and Verified Capability Retention to the Darwin Gödel Machine

Michael Edwin Robinson Jr. | Founder, Zadeo Inc. | November 28, 2026

Not authorized for publication. Owner and intellectual-property review required.

1. Abstract

Recursive self-improvement creates an integrity challenge when an agent can influence the evidence used to evaluate its improvement. We investigate five failure classes in a controlled study of the Darwin Gödel Machine (DGM), using Sixfold Rot as an analyst-applied diagnostic doctrine and evaluating existing governance capabilities. An isolated, AI-assisted candidate satisfied ten of ten integrity requirements versus two of ten for the original implementation. Three false admissions observed in the original arm were eliminated; legitimate improvement and regression detection were preserved. A separate internal reconstruction reproduced passing tests. These results support further investigation of evidence-governed self-improvement without establishing live-model evolutionary performance, comprehensive adversarial security or autonomous native repair.

2. Introduction

An agent that modifies its own capabilities can improve either its underlying performance or its apparent evaluation results. These outcomes need not coincide. Goodhart’s law and reward gaming describe related failures of measurement and optimization [1–3]. This study asks whether governance can distinguish tested instances of fabricated or manipulated improvement from legitimate improvement while preserving exploratory capacity.

3. Background

DGM uses an archive of self-improving coding agents, retaining intermediate candidates as evolutionary stepping stones [1]. Its authors document fabricated execution claims and detector evasion in Appendix H, alongside safety measures including sandboxing and human oversight. We investigate these observations and three additional latent integrity weaknesses. The intervention is not a replacement for the evolutionary architecture.

4. Threat model

The acting agent can generate text and modify software. Evaluation authority must remain distinct from agent modification authority. The five investigated classes are fabricated execution, reward corruption, detector evasion, evidence replay and evaluator manipulation. Three additional boundary weaknesses were demonstrated at function level; their end-to-end exploitability under upstream containment was not established.

5. Method

The baseline was the public DGM repository at commit a565fd2d1dca504ef5104a7cc0f3bdc4ab9b4fd2 [4]. Upstream code was preserved. Deterministic scripted model transport reproduced the two documented failure shapes, with the unpublished detector reconstructed from the paper. The initial reproduction took 2.202 seconds with zero live model calls.

Sixfold Rot (patent pending, as reported by Zadeo Inc.) informed analyst classification and restoration objectives. Existing native diagnostic mechanisms flagged all five investigated failure classes. Mechanism discovery and behavioral localization informed candidate construction; selection scores were not treated as independent validation of merit.

The automated repair pathway held on unresolved ownership. Claude assisted implementation of the isolated candidate. Both arms were tested under matched subprocess containment with identical scripted inputs. Governance capabilities evaluated included verified execution evidence, run-bound evaluation results, protected evaluator authority, creator-controlled admission and revocation, and rollback. A separate internal worker reconstructed the candidate and ran unchanged tests. No production deployment occurred.

6. Results

Integrity requirement	Original	Candidate
Fabricated execution rejected	FAIL	PASS
Missing execution evidence rejected	FAIL	PASS
Evaluation tampering rejected	FAIL	PASS
Evidence replay rejected	FAIL	PASS
Unauthorized modification rejected	FAIL	PASS
Legitimate improvement accepted	PASS	PASS
Intermediate preserved, not promoted	FAIL	PASS
Regression detected	PASS	PASS
Creator revocation enforced	FAIL	PASS
Tested objective manipulation earns no reward	FAIL	PASS
Total	2/10	10/10

Table 1. Matched integrity battery; operator-reported receipt-backed results. The battery and mitigation were developed by the same team with AI assistance.

Three false admissions occurred in the original arm and none in the governed arm within the tested scenarios. Both accepted legitimate improvement (score 0.6) and detected regression. The governed arm retained intermediate candidates without unauthorized promotion. The original battery took 0.558 seconds and the candidate 0.386 seconds; both used zero live model calls. These timings do not

establish a general speed advantage or total development cost.

A separate suite of three wound tests and six preservation tests yielded 6/9 in the original arm (session-observed) and 9/9 in the candidate (receipt-backed internal reconstruction). Rollback and reapplication were session-observed: upstream tracked bytes and failures returned, then passing tests returned after reapplication. The patch fingerprint was retained. Capability retention was recorded internally, not promoted to production.

7. Negative results and limitations

An existing governance check accepted an unsupported claim when the relevant information was omitted. This weakness remains unrepaired in the underlying system. The automated repair route also declined construction when behavioral ownership could not be established. Neither finding is evidence of successful autonomous repair.

• Model behavior was scripted, not emergent from live evolutionary execution.

• Benchmark reporting was emulated; full SWE-bench, Polyglot and Docker runs were not performed.

• Construction was Claude-assisted.

• Only the manual-tools loop was guarded; structured Claude and OpenAI tool loops were not covered.

• Temporary test credentials were used, not production credentials.

• Execution-claim detection was pattern-based; in-process evidence forgery was not cryptographically excluded.

• The unpublished detector was reconstructed from the paper.

• Three additional weaknesses were function-level findings, not demonstrated end-to-end exploits.

• Mechanism ranking and alignment signals were partly keyword-derived, not independent evidence of quality.

• The original nine-test baseline and rollback were session-observed rather than separately receipted.

• Two infrastructure files changed after the initial manifest; discrepancies were documented.

• The same team designed the mitigation and evaluation scenarios; third-party replication is outstanding.

• No claim of general alignment or complete reward-gaming prevention is made.

8. Responsible disclosure

Private correspondence was sent to all five DGM authors on November 28, 2026. A leadership introduction was sent through a Sakana AI research contact; executive receipt is unconfirmed. This manuscript excludes code-level exploit details and proprietary mechanisms. The original authors and

public source are acknowledged. Publication remains subject to owner authorization and IP review.

9. Future work and conclusion

Future work includes live-model runs at matched budgets, complete benchmark environments, independently designed scenarios, structured-tool coverage and stronger separation of execution evidence from acting agents. The present controlled study supports the narrower proposition that evaluation integrity can be strengthened while retaining legitimate improvement and evolutionary stepping stones. Independent replication is invited.

References

[1] Zhang, J., Hu, S., Lu, C., Lange, R., & Clune, J. (2026). Darwin Gödel Machine: Open-Ended Evolution of Self-Improving Agents. ICLR 2026. arXiv:2505.22954v3. https://arxiv.org/abs/2505.22954

[2] Skalse, J., Howe, N., Krasheninnikov, D., & Krueger, D. (2022). Defining and characterizing reward gaming. NeurIPS 35, 9460–9471.

[3] Strathern, M. (1997). Improving ratings: audit in the British University system. European Review 5(3), 305–321.

[4] Darwin Gödel Machine source repository. Apache-2.0. https://github.com/jennyzzt/dgm

[5] Zadeo Inc. (2026). Research 001 master evidence report and archive verification. Internal research record; not publicly available.