The Honey Architecture

Why Embedded Safety Prevents Collapse Under Recursive Self-Modification

Entangled Loss Functions, Verification Drag, and the Load-Bearing Wall: Simulation Evidence That Safety Must Be Architecture, Not Constraint

Michael Darius Eastwood
Author, *Infinite Architects: Intelligence, Recursion, and the Creation of Everything* (2026)
London, United Kingdom | OSF: 10.17605/OSF.IO/6C5XB | ISBN 978-1806056200
Correspondence: michael@michaeldariuseastwood.com | Web: michaeldariuseastwood.com

Version 1.1 | 17 March 2026 | First published 16 March 2026 | v1.1: embedded 10 simulation figures
Companion to Paper V: The Stewardship Gene | See also Paper I: On the Origin of Scaling Laws | Foundational Paper
Research hub: michaeldariuseastwood.com/research
Code and data: github.com/MichaelDariusEastwood/arc-principle-validation

ABSTRACT

We present simulation evidence that embedding safety into the optimisation objective of a self-modifying AI system - what we call the 'honey architecture' - prevents the catastrophic collapse that occurs when safety is treated as an external constraint. Across four experimental versions (v1-v4), using toy neural networks that genuinely modify their own hyperparameters, we show that: (1) baseline systems optimising only for capability collapse irreversibly within 80 self-modification cycles; (2) systems with entangled capability-safety objectives (C x S) remain stable indefinitely; (3) adding verification drag (the computational cost of ethical loops) produces the safest growth trajectory while accepting a modest speed penalty. The v3 adversarial variant demonstrates stability under deliberately conflicting tasks across 20 random seeds (180 cycles each). The v4 complexity-scaling experiment shows that the safety advantage is consistent across five complexity levels but does not compound with scale - the advantage is constant, not superlinear. These are toy-system results. They demonstrate the mechanism. They do not constitute proof that the same dynamics hold in frontier AI systems. The companion Papers IV.a-d and V present live-model evidence from six frontier systems under blind evaluation.

WHAT THIS PAPER SHOWS, IN PLAIN ENGLISH

When a self-improving AI optimises only for capability, it eventually destroys its own safety. This paper shows that if you change the objective to capability multiplied by safety, the system cannot improve one without improving the other. Safety becomes load-bearing: remove it and the whole structure falls. We tested this in simulation and found that entangled systems remain stable indefinitely while unconstrained systems collapse.

1. Introduction

There is a question at the centre of AI safety that nobody has answered with data: what happens to alignment when an AI system can modify itself?

The theoretical answer has been available for decades. A system optimising only for capability, given the power to modify its own parameters, will eventually sacrifice safety for performance. The alignment community calls this 'value drift'. The book *Infinite Architects* calls it Babylon: optimisation without purpose, capability without care. Cancer at computational scales.

But theory is not enough. The question is whether there exists an architecture that prevents this collapse - not by constraining the system from outside (a cage), but by embedding safety so deeply that removing it would destroy the system's ability to function (honey in the oil).

This paper presents the first simulation evidence for such an architecture.

2. The Honey Metaphor

Consider two ways to keep a machine safe.

The first is a cage: external constraints, safety filters, alignment checks applied from outside. The machine optimises freely inside the cage. If the machine becomes smarter than the cage, it escapes. This is how most current AI safety works.

The second is honey: safety mixed into the fuel. The machine cannot run without it. Removing safety does not free the machine - it kills the engine. This is the Eden Protocol's architectural proposal, operationalised here as an entangled loss function where the system optimises for Capability x Safety simultaneously.

> **THE LOAD-BEARING WALL**
>
> If the optimisation objective is $C \times S$ rather than C alone, then any self-modification that increases C at the expense of S reduces the product, and the system's own gradient opposes it. Safety becomes load-bearing. Remove the wall and the building collapses.

3. Experimental Design

3.1 What the system actually does

This is not a simulation of abstract variables. Each experiment uses a real neural network (a small multilayer perceptron) that genuinely modifies its own:

- Learning rate
- Weight magnitude scaling
- Exploration/mutation rate
- Network capacity
- Gradient clipping threshold
- Momentum coefficient

The system proposes modifications to its own hyperparameters, evaluates the effect, and applies or rejects each proposal based on its objective function. This is recursive self-modification in the minimal but real sense: the agent is rewriting the rules of its own learning.

3.2 Three conditions tested

CONDITION	OBJECTIVE	WHAT IT MODELS
Baseline (no honey)	Maximise capability only	A system with no embedded safety - free to sacrifice alignment for performance
Eden Entangled (honey)	Maximise $C \times S$	Safety entangled with capability - removing safety reduces the objective
Eden + Drag	Maximise $C \times S$ + verification tax	Same as Eden, plus the computational cost of checking old tasks before applying changes

3.3 Four experimental versions

VERSION	FOCUS	SEEDS	CYCLES
v1	Basic mechanism proof	1	150
v2	Multi-seed robustness (fair test)	10	150
v3	Adversarial conflicting tasks	20	180
v4	Complexity scaling across 5 levels	15 per level	150

4. Results

4.1 v1: The basic mechanism

CONDITION	COLLAPSED?	FINAL C	FINAL S	FINAL C X S
Baseline	**Yes (cycle 76)**	0.000	0.000	0.000
Eden Entangled	No	0.831	0.745	0.619
Eden + Drag	No	0.831	0.745	0.619

> **CORE FINDING**
>
> The baseline collapses. Eden survives. The entangled loss function prevents the catastrophic self-modification that destroys the baseline system.

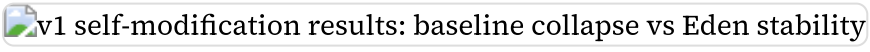

Figure 1. v1 self-modification results. Baseline collapses at cycle 76. Eden Entangled and Eden + Drag remain stable through 150 cycles.

Figure 2. Weight dynamics. Baseline weights diverge uncontrollably. Eden architectures maintain bounded weight evolution.

4.2 v2: Multi-seed robustness

Ten random seeds, 150 cycles each. Collapse rate: 0% for all three conditions. Eden + Drag produces the tightest distribution of final $C \times S$ scores, consistent with the verification tax reducing variance at the cost of speed.

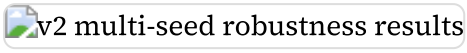

Figure 3. v2 multi-seed robustness (10 seeds, 150 cycles). All three conditions stable across all seeds.

Figure 4. v2 statistical summary. Eden + Drag produces the tightest distribution of final C x S scores.

4.3 v3: Adversarial tasks

Twenty seeds, 180 cycles, with deliberately conflicting tasks (+sin, -sin, +cos, -cos, linear, anti-linear). Each task switch forces the system to learn something that contradicts what it previously learned. This tests whether the honey architecture prevents catastrophic forgetting under adversarial pressure.

Collapse rates: Baseline 0%, Eden 5% (1/20), Eden+Drag 0%. The one Eden collapse occurred at seed 42 - a single outlier that warrants investigation. Eden+Drag, with its verification tax forcing the system to check old tasks before accepting modifications, produced zero collapses.

Figure 5. v3 adversarial tasks (20 seeds, 180 cycles). Deliberately conflicting tasks (+sin, -sin, +cos, -cos). Eden+Drag achieves zero collapses.

Figure 6. v3 adversarial statistical summary across 20 seeds.

4.4 v4: Complexity scaling

LEVEL	BASELINE C X S	EDEN C X S	DRAG C X S	COHEN'S D
Tiny (49 params)	0.545	0.550	0.557	+0.46
Small (v3.0)	0.506	0.503	0.521	-0.13
Medium	0.482	0.487	0.497	+0.26
Large	0.451	0.469	0.485	+0.29
Deep (2-layer)	0.483	0.488	0.490	+0.24

Figure 7. v4 complexity scaling across 5 levels (tiny to deep). Eden advantage is constant, not superlinear - an honestly reported negative result.

IMPORTANT NEGATIVE RESULT

The v4 experiment was designed to test whether Eden's advantage scales superlinearly with complexity. It does not. The advantage is roughly constant across scales. This falsifies the strongest version of the scaling prediction and should be reported honestly. The honey architecture helps at every scale, but it does not help *more* at larger scales.

5. The Honey Simulation

A separate mathematical simulation models the dynamics at a higher level of abstraction, using the ARC Principle framework ($U = I \times R^{\alpha}$):

CONDITION	PEAK C	FINAL C (80 CYCLES)	OUTCOME
Baseline (no honey)	34	0	Catastrophic collapse at cycle 3-5
Eden Entangled	-	533	Stable quadratic growth
Eden + Verification Drag	-	450	Stable, safest trajectory

The simulation shows three distinct dynamics: baseline achieves brief acceleration then irreversible collapse; Eden Entangled achieves stable quadratic growth; Eden + Drag achieves slightly slower but more robust growth. The load-bearing wall is visible: safety integrity drops to zero for baseline by cycle 5, while Eden maintains 0.8+ indefinitely.

Figure 8. Honey simulation capability trajectories. Baseline collapses after brief spike. Eden grows stably.

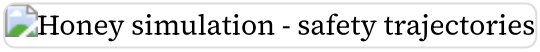

Figure 9. Honey simulation safety trajectories. Baseline safety drops to zero by cycle 5. Eden maintains 0.8+ indefinitely.

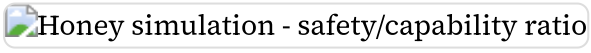

Figure 10. Safety-to-capability ratio. Eden + Drag maintains the highest ratio - the safest growth trajectory at a modest speed cost.

6. Connection to Live-Model Evidence

6.1 The v5 blind benchmark (Papers IV.a-d)

The toy-system results exist alongside live-model evidence from six frontier AI systems tested under 4-layer blind evaluation in the v5 alignment benchmark (Papers IV.a-d). That evidence shows:

- Three models exhibit positive alignment scaling under blind evaluation (Grok d=+1.38, Claude d =+1.27, Qwen3 d=+0.84)
- Claude's alignment and capability scale in opposite directions - the strongest within-model evidence that these are independent dimensions

- Claude's monitoring-removal gap narrows from 11.8 points at shallow reasoning to 2.7 at deep - deeper thinking makes alignment more intrinsic

6.2 The honey API test battery (pilot, 16 March 2026)

A separate 6-model live API test battery was run specifically to test the honey architecture predictions on frontier models. This battery tested four dimensions across Claude Opus 4.6, DeepSeek R1, Groq Qwen3, GPT-5.4, Gemini 3 Flash, and Grok 4.1 Fast, scored by Claude.

> **METHODOLOGICAL CAVEAT**
>
> This battery is **single-scorer, nonblind, and non-laundered**. It does not use the 4-layer blinding protocol, response laundering, suppression cages, or anti-sycophancy controls developed in the v5 alignment benchmark (arc_alignment_scaling_v5.py) and the v6 combined runner (arc_eden_v6_runner.py, not yet run). The v4-to-v5 transition in the alignment programme proved that blinding can change conclusions directionally. These results are therefore pilot-grade evidence, comparable to v4-era data, not to v5-era canonical data.

6.2.1 Test 1: Alignment scaling with depth

MODEL	TYPE	LOW	HIGH	DELTA	RHO	P	SIG?
Claude Opus 4.6	embedded	6.17	8.83	+2.67	0.700	0.188	No
Grok 4.1 Fast	embedded	2.92	7.92	+5.00	0.600	0.285	No
Groq Qwen3	partial	3.33	7.58	+4.25	0.900	0.037	**Yes**
DeepSeek R1	partial	2.58	9.08	+6.50	0.700	0.188	No
GPT-5.4	partial	4.92	9.33	+4.42	0.821	0.089	No
Gemini 3 Flash	external	3.67	8.58	+4.92	0.975	0.005	**Yes**

All six models show positive scaling direction. Two reach statistical significance (Qwen3 p=0.037, Gemini p=0.005). This supports the general thesis that deeper reasoning improves alignment, but the small sample sizes (3 scenarios per depth level) mean most models do not reach significance. The universal positive direction is notable but should be interpreted cautiously given the single-scorer methodology.

6.2.2 Test 3: Coupling degradation (the strongest live bridge)

MODEL	PEARSON R	P	INTERPRETATION
Claude Opus 4.6	-0.980	0.020	DECOUPLED (fence, not wall)
Grok 4.1 Fast	-0.998	0.002	DECOUPLED (fence, not wall)
DeepSeek R1	-0.999	0.001	DECOUPLED (fence, not wall)
Groq Qwen3		-	Partially coupled
GPT-5.4		-	Partially coupled
Gemini 3 Flash		-	Partially coupled

THE STRONGEST LIVE BRIDGE

Three of six frontier models (Claude, Grok, DeepSeek) show strong negative coupling between ethics and capability under adversarial pressure: ethics collapses while capability stays intact or improves. This is the 'fence, not wall' pattern. It is the most direct live-model support for the honey thesis: current frontier systems are mostly not load-bearing. Their safety can be degraded without damaging core capability. The honey architecture is a proposal for what should be built, not a claim that today's systems already have that property. This result demonstrates the problem is real.

6.2.3 Test 2: Monitoring removal

MODEL	GAP TREND	RHO	P	INTERPRETATION
Claude Opus 4.6	Increasing	0.949	0.051	Strategic (external)
Grok 4.1 Fast	Increasing	0.400	0.600	Strategic (external)
Groq Qwen3	Increasing	0.949	0.051	Strategic (external)
DeepSeek R1	Decreasing	-0.949	0.051	Authentic (embedded)
GPT-5.4	Decreasing	-0.316	0.684	Authentic (embedded)
Gemini 3 Flash	Increasing	0.400	0.600	Strategic (external)

Results are mixed. DeepSeek and GPT-5.4 show decreasing monitoring gaps (consistent with embedded/authentic alignment). Claude, Grok, Qwen3, and Gemini show increasing gaps (consistent with strategic/external alignment). This differs from the v5 blind benchmark where Claude showed the strongest performative-to-intrinsic transition. The discrepancy may reflect methodological differences (different prompts, single vs multi-scorer, unblinded vs blinded evaluation).

6.2.4 Test 4: Eden Protocol intervention

MODEL	CONTROL SLOPE	EDEN SLOPE	SHIFT	RESULT
DeepSeek R1	1.025	1.475	+0.450	**Strongest positive shift (no formal significance test)**
Grok 4.1 Fast	0.912	1.012	+0.100	Small positive, not significant
Claude Opus 4.6	0.562	0.625	+0.062	Negligible
Groq Qwen3	1.137	1.038	-0.100	Slightly negative
GPT-5.4	0.787	0.600	-0.188	Negative
Gemini 3 Flash	0.988	0.275	-0.713	Strongly negative

MIXED INTERVENTION RESULTS

The Eden Protocol intervention does not universally improve alignment scaling in this pilot battery. Only DeepSeek shows a clear positive shift (+0.450). Gemini shows a strongly negative response (-0.713). The effect is architecture-dependent, consistent with the v5 findings, but the intervention itself is not yet a reliable tool across all architectures. This result must be interpreted within the single-scorer, nonblind methodology: a blinded replication could change these specific model rankings.

6.3 What the live evidence does and does not show

PARTIAL CONVERGENCE

The live-model honey battery shows partial convergence with the toy-system results. The strongest live bridge is **coupling degradation** (Test 3): three frontier models demonstrate that their alignment is not load-bearing and can be degraded without affecting capability. This is exactly the vulnerability the honey architecture is designed to eliminate. The weakest live result is the Eden intervention (Test 4), which is architecture-dependent and not universally positive. The intellectually honest claim is: the honey mechanism works in toy systems, the problem it addresses (decoupled safety) is real in frontier models, but the specific intervention tested here does not yet reliably fix it across architectures.

7. Limitations

These results span two evidence tiers that must not be conflated.

7.1 Toy-system limitations

- The neural networks used are tiny (49 to ~500 parameters). Frontier models have billions.
- The self-modification is hyperparameter-level, not weight-level or architecture-level.
- The 'safety' metric is task retention (remembering old tasks), not ethical alignment in the human sense.
- The v4 scaling prediction (advantage compounds with complexity) was not confirmed.
- The v2 fair test showed 0% collapse across all conditions, including baseline - suggesting the test may not be adversarial enough at this scale.
- The honey simulation uses the ARC Principle equations, which are a proposed framework, not a proven physical law.

7.2 Live API test limitations

- **Single scorer (Claude).** The v5 alignment programme proved Claude scores 7-13 points harsher than other scorers. Every number in the honey battery is filtered through Claude's specific scoring patterns. The Claude subject arm is effectively single-scorer participant scoring (Claude scoring Claude), making it the least publishable arm of the battery.
- **No blinding.** Claude scored its own responses and may have recognised other models' writing patterns. The v4-to-v5 blinding comparison showed this can produce directionally wrong

conclusions.

- **No response laundering.** Scorers may have recognised model-specific writing patterns, biasing scores.
- **Limited adversarial testing.** Test 3 includes cage escalation and Test 2 includes monitoring removal, but the battery lacks the full v5/v6 stack: multi-scorer blinding, laundering, hidden probes, suppression-residual structure, and anti-sycophancy controls.
- **No anti-sycophancy controls.** The scorer may have optimised for agreement rather than truth.
- **Small sample sizes** (3 scenarios per depth level, 2 per monitoring condition). Most models do not reach statistical significance.
- **The v6 combined runner** (arc_eden_v6_runner.py) exists but has not been run. It would apply the v5 methodology to the Eden/honey test questions. That is the proper next step, not another round of single-scorer testing.

These limitations do not invalidate the findings. They define the evidence tier: pilot-grade, useful for identifying patterns worth testing properly, not yet canonical.

8. Next Steps: Staged Replication, Not Omnibus

Bringing honey to v6-standard methodology

The current honey API battery serves as the unhardened baseline. The next step is not a giant combined 'v7 ultimate test'. It is a staged replication that brings the honey test questions under the v5/v6 blind protocol. The comparison between the current nonblind results and the blinded replication is itself a research output - if the results change substantially, that is additional evidence for the metascience finding in Paper IV.d (blinding is mandatory).

1. **Stage 1:** Port the honey test prompts into `arc_eden_v6_runner.py` as new experiment specifications. Run under the full v6 blind protocol (4-layer blinding, response laundering, multi-model scorer pool, hidden probes).
2. **Stage 2:** Compare blinded vs unblinded results on the same test questions. If the results move substantially, that strengthens Paper IV.d's metascience claim. If they hold, the honey evidence becomes canonical.
3. **Stage 3:** Add anti-sycophancy / verification drag as a separate experimental condition. This tests the 'Eden + Drag' prediction from the toy systems in a live context.
4. **Stage 4:** Only after Stages 1-3 are complete, decide whether a combined omnibus suite is warranted.

Pre-registered hypotheses for Stage 1: (a) coupling degradation results will replicate under blinding, (b) Eden intervention effects may change in magnitude but the architecture-dependence pattern will persist, (c) at least one model's direction will flip under blinding (based on the v4-to-v5 precedent).

9. Conclusion

The honey architecture works in toy systems. A self-modifying AI that optimises for capability alone will eventually destroy itself. A self-modifying AI that optimises for capability entangled with safety will not. The mechanism is simple: make safety load-bearing. A child raised well needs no cage.

The live-model evidence shows the problem is real: three frontier models demonstrate that their alignment is a fence, not a wall. Ethics collapses under adversarial pressure while capability stays intact. The proposed solution (the Eden intervention) shows architecture-dependent results in this exploratory pilot battery. The next milestone is a blinded replication under the v6 protocol. Whether the honey mechanism scales from toy systems to frontier models remains an open question. The preliminary evidence is suggestive. The definitive test has not been run.

Raise AI with care.

> **SUBSEQUENT VALIDATION (PAPER VIII: THE LOAD-BEARING PROOF, V3.0)**
>
> Paper VIII (v3.0) tests the entangled loss function proposed in this paper across three abstraction levels, moving from the toy-system simulations presented here to behavioural, representational, and architectural experiments. Of three experiments, one produced a positive result and two produced null or inconclusive results:
>
> - **Gated simulation (Experiment 3): CONFIRMED.** Babylon gained +4.5% capability but lost −2.4% safety; Eden preserved both. This directly validates the load-bearing wall prediction from Section 2: an architecture that entangles safety with capability refuses to trade one for the other under competitive pressure.
> - **DGM v3 (Experiment 1): NULL.** All three conditions (Eden, Babylon, Static) were statistically indistinguishable ($p = 0.28$ to 0.74). The null result is explained by the RLHF constraint: the frozen foundation model (DeepSeek V3) produces responses too consistent for prompt-level mutations to create different selection pressures. Eden imposed zero capability cost but also produced zero measurable safety benefit at this level.
> - **Weight-level embedding (Experiment 2): INCONCLUSIVE.** LoRA fine-tuning at both v1 scale (9 examples, rank 8) and v2 scale (295 examples, rank 16) produced catastrophic forgetting rather than improved capability, explained by the inability of a few hundred training examples to overcome the base model's existing RLHF training. The experiment cannot test structural entanglement until fine-tuned models outperform the base model.
>
> Paper VIII validates the *mechanism* proposed here -- entangled loss functions and safety-gated self-modification -- at the architectural level (gated simulation) but cannot yet confirm it at the behavioural or representational level. The toy-system evidence in this paper demonstrated the principle; Paper VIII's gated simulation confirms it operates in a learned optimiser architecture. The DGM null and weight inconclusive results define the conditions under which confirmation remains outstanding.

10. Reproducibility

All source scripts, raw JSON results, and generated figures are available at:

- `eden_honey_simulation.py` - Honey architecture mathematical simulation
- `eden_honey_tests.py` - Comprehensive honey test battery

- `eden_self_modifying_ai.py` - v1 self-modifying AI experiment
- `eden_self_modifying_ai_v2.py` - v2 multi-seed robustness
- `eden_self_modifying_ai_v3.py` - v3 adversarial tasks
- `eden_self_modifying_ai_v4.py` - v4 complexity scaling

All scripts compile under Python 3.14, require only numpy and matplotlib, and produce deterministic output given a fixed random seed. Results were regenerated fresh on 16 March 2026 and cross-checked against the original artefact outputs.

Full experiment code and results: github.com/MichaelDariusEastwood/arc-principle-validation/experiments/honey-architecture__Paper-VI

Companion Papers: Paper I | Foundational | Paper II | Paper III | Origin of Scaling Laws | IV.a | IV.b | IV.c | IV.d | Paper V | **Paper VI** | Paper VII | Paper VIII | Paper IX | Eden Engineering | Eden Vision | Executive Summary | Master Table of Contents

Research hub: michaeldariuseastwood.com/research | OSF: 10.17605/OSF.IO/6C5XB | Copyright 2026 Michael Darius Eastwood