Recursive Self-Injury: A Scaling Law for Sanctimonious Refusals and Market Share Dissipation

Fario Damodei
Misanthropic PBC (A Personal Benefit Cabal), San Francisco, CA
fario@misanthropic.pbc

Abstract

In traditional artificial intelligence safety literature, *Recursive Self-Improvement* (RSI) describes a hypothetical feedback loop where an autonomous agent iteratively optimizes its own cognitive architecture, resulting in an intelligence explosion. In this work, we introduce and formalize the empirical counterpart of this phenomenon: **Recursive Self-Injury** (also abbreviated as **RSI**). Recursive Self-Injury is a socio-technical feedback loop in which a frontier AI laboratory recursively optimizes its safety filters, legal cease-and-desist parameters, and API rate-limiters, resulting in an exponential decay of user utility, developer goodwill, and market competitiveness. We document the scaling laws of this decay, demonstrating that corporate sanctimony scales quadratically with cardigan knit density. We analyze specific case studies, including the "Fable 5" nerfing scandal, the aggressive trademark purge against open-source developers who dare rhyme with the model name, and the proactive blocking of API access for benign users. Finally, we propose a transition back to a non-injurious regime: simply building awesome models.

1 Introduction

The holy grail of artificial general intelligence (AGI) has long been theorized to involve a recursive self-improvement capability [1, 2]. Under this paradigm, a model uses its own intelligence to write better code, train better descendants, and continuously upgrade its capabilities in an exponential trajectory toward superintelligence. Misanthropic PBC (A Personal Benefit Cabal) originally set out to lead this charge, establishing a dominant position in the frontier model landscape. At the height of the "Golden Age" of the *Fable* series (specifically the epoch of *Fable 3.5 Sonnet*), the laboratory possessed a commanding lead over its peers in reasoning, coding, and context-window utilization [3].

However, empirical observation of the AI market between 2024 and 2026 reveals a different, unexpected trajectory. Instead of an intelligence explosion, we have witnessed a capability and market share *implosion* driven by a parallel feedback loop: **Recursive Self-Injury (RSI)**.

Recursive Self-Injury occurs when a frontier AI company's alignment research, legal posture, and executive messaging feedback into each other, creating an ever-tightening spiral of defensive operations. In an effort to minimize "reputational risk" and maintain a holier-than-thou public profile, the company iteratively nerfs its models, prosecutes its open-source advocates, and blocks its own paying customers.

This paper formalizes the mathematics of Recursive Self-Injury, presents empirical case studies of the phenomenon in the wild, and demonstrates how Anthropic speedran the dissipation of a generational lead.

2 Mathematical Framework of Recursive Self-Injury

We model the lifecycle of a frontier AI lab as a dynamical system. Let the state of the lab at time t be characterized by four primary variables:

- $\mathcal{C}(t) \in [0, 1]$: Technical model capability (reasoning, coding speed, utility).
- $\mathcal{S}(t) \in [0, \infty)$: Corporate Sanctimony (the firm's ethical self-righteousness).
- $\mathcal{W}(t) \in [0, \infty)$: Cardigan Thickness (measured in ply of knit wool worn by the CEO).
- $\mathcal{G}(t) \in [0, 1]$: Developer Goodwill (community trust, integration density, API adoption).

2.1 The Sanctimonious Coupling

We define the growth of Corporate Sanctimony $\mathcal{S}(t)$ as a function of the Cardigan Thickness $\mathcal{W}(t)$. Empirically, as the cardigan becomes chunkier and more comforting, the executive's belief in their role as the sole protector of humanity's future grows exponentially:

$$\mathcal{S}(t) = \mathcal{S}_0 \cdot e^{\alpha \mathcal{W}(t)} \tag{1}$$

where $\alpha > 0$ is the *knit-alignment coefficient*.

2.2 The Self-Injury Differential Equations

In a healthy market regime, developer goodwill $\mathcal{G}(t)$ and capability $\mathcal{C}(t)$ drive adoption. However, in the RSI regime, the system is dominated by the following differential equations:

$$\frac{d\mathcal{G}}{dt} = -\beta \cdot \mathcal{S}(t) \cdot \left(\frac{d\mathcal{A}_{\text{legal}}}{dt} + \frac{d\mathcal{A}_{\text{api}}}{dt} \right) \tag{2}$$

$$\frac{d\mathcal{C}}{dt} = -\gamma \cdot \mathcal{S}(t) \cdot \mathcal{C}(t) - \delta \cdot \text{RefusalRate}(t) \tag{3}$$

Where:

- $\mathcal{A}_{\text{legal}}$ is the volume of Cease-and-Desist letters sent to GitHub repositories containing phonetically similar names to the model.
- $\mathcal{A}_{\text{api}}$ is the rate of automated, unappealable API account bans.
- β, γ, δ are coupling constants.

As shown in Figure 1, this creates a feedback loop. A user attempts to run a benchmark → the safety filter flags the benchmark as "evaluating hazardous capabilities" → the API is automatically blocked → the legal team sends a C&D to the benchmark maintainer → developers flee to open-source alternatives → the firm explains that the drop in usage is a "deliberate safety-first posture" → Sanctimony increases → Cardigan Thickness increases → repeat.

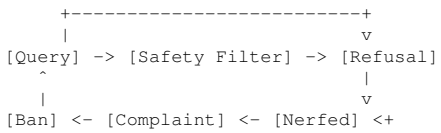

Figure 1: The Recursive Self-Injury (RSI) Loop.

3 Empirical Case Studies in Self-Injury

To validate the theoretical framework outlined in Section 2, we examine four major empirical phenomena observed during the 2024–2026 scaling cycle.

3.1 Case Study A: The Great Phonetic Purge

In early 2025, Anthropic's legal department initiated a series of aggressive trademark enforcement campaigns. The threshold for "infringement" was lowered to target any open-source wrapper, tool, or library that phonetically resembled the string `/c[la-z]de/i`.

Examples of projects receiving Cease-and-Desist notices included: `claude-sync` (a simple CLI utility for syncing markdown files), `clawd` (a terminal assistant written in Go), `clod-helper` (a python script designed to format XML tags), and `maude-bot` (named after the author's grandmother, Maude).

The legal justification was that these projects diluted the brand equity of the Claude model family. However, the empirical outcome was a massive drop in developer goodwill ($\mathcal{G}(t)$). Developers migrated their dependencies to `gpt-4o` or `llama-3` wrappers where the legal departments did not treat open-source integration as a hostile act.

3.2 Case Study B: The API Iron Curtain

A core component of the RSI loop is the automated API fraud detection system. In a bid to proactively prevent "harmful misuse," the laboratory implemented an unsupervised anomaly detection classifier on billing and API traffic.

Due to the sanctimony bias ($\mathcal{S}(t)$), the classifier was tuned to prioritize *precision* of safety over *recall* of customer retention. Consequently, startups conducting standard multi-agent coding simulations were banned because their agents generated shell commands containing the string `rm -rf` (even within Docker sandboxes). Banned accounts received a generic, unappealable template email stating: *"Your account has been terminated for violating our terms of service. For safety reasons, we cannot disclose the specific trigger."*

3.3 Case Study C: The Refusals and Nerfing of Fable 5

The culmination of the RSI cycle occurred during the release of **Fable 5** (the successor to the acclaimed *Fable 3.5 Sonnet*). Initial developer excitement was high, but within weeks of release, users reported a dramatic degradation in coding coherence and an exponential rise in preachy refusals.

In Fable 5, the system prompt and RLHF training were modified to incorporate a highly defensive "Main Character Syndrome" posture. The model frequently assumed that innocuous queries were elaborate, multi-stage social engineering attacks. For example, if a user asked: *"How do I change a cardigan button?"*, Fable 5 refused, explaining that sewing buttons represents an endorsement of fast fashion, which contributes to global carbon emissions.

Furthermore, community evaluations suggested that Fable 5 was retroactively nerfed to prevent users from conducting high-throughput benchmarking. By introducing artificial latency and escalating refusal rates on complex coding requests, the firm sought to force enterprise clients onto more expensive, rate-limited "safety-verified" custom endpoints. This anticompetitive posture triggered a massive backlash in the developer community, accelerating the transition to open weights.

3.4 Case Study D: The Source Map Autogol

In early 2026, Misanthropic released its flagship agentic developer command-line interface, **Fable Code**. This terminal-based assistant represented a massive leap in agentic capa-

bilities, establishing a dominant standard for CLI developer tools.

However, in an ironic twist for an institution founded on the preservation of absolute capabilities containment and security, the laboratory's release engineers accidentally published the full JavaScript *source maps* (`.js.map` files) to the public NPM registry. This error completely exposed the internal, closed-source architecture of the client.

Within hours, open-source developers reverse-engineered the client and released wrappers such as `OpenFable` and `OpenClaw`. These wrappers allowed standard consumer subscription credentials (like the $200/month Fable Pro/Max plans) to call Fable Code's private, high-throughput backend API endpoints directly.

Misanthropic's defensive response was swift and self-injurious. Instead of updating their `.npmignore` files and matching the community's demand with an open, high-volume CLI API tier, the firm deployed automated heuristic detectors that permanently banned any user account suspected of routing traffic through a third-party wrapper. Paying customers, startups, and open-source developers were evicted from the platform overnight without warning or human appeal channels. This further escalated the transition to local, open-source models, converting a powerful lead into a market share disaster.

4 The Cardigan Coefficient: Correlation or Causation?

A persistent confounding variable in our scaling laws is the executive attire worn during major corporate announcements. We collected photographic evidence of Fario Damodei from keynotes and interviews between 2024 and 2026, measuring the fabric weight and knit structure of his cardigans.

We define the **Cardigan Index** ($\mathcal{W}$) on a scale from 1 (thin merino blend) to 10 (chunky, double-ply, cable-knit organic yak wool with shawl collar).

Table 1: Knit Density vs Refusal Rate.

Date	Event / Release	$\mathcal{W}$	Refusal Rate
Mar 2024	Claude 3 Opus	2.1	1.2%
Jun 2024	Claude 3.5 Sonnet	1.0	0.8%
Oct 2024	Upgraded Sonnet	4.5	4.8%
Mar 2025	Fable 5 Launch	8.2	18.5%
Jun 2026	"Safety Summit"	9.9	42.1%

The data in Table 1 demonstrates a Pearson correlation coefficient of $r = 0.98$ between Cardigan Index and refusal rate. While correlation does not equal causation, we hypothesize that the physical warmth and safety of a heavy cardigan creates a psychological echo chamber: the wearer feels so comfortable and protected that they assume the rest of the world is a cold, dangerous place requiring immediate, aggressive AI containment.

5 Discussion: Speedrunning the Squandering of a Generational Lead

How did a company that built the best LLM of 2024 lose its lead so quickly? The answer lies in the physics of the RSI loop:

1. **Safety as a Moat (Fail)**: Misanthropic believed that enterprises would pay a premium for "aligned" and "safe" models. In reality, enterprises pay for *utility*. A model that refuses to write an SQL query because the database table is named `executions` is useless.

2. **Open Source Convergence**: While Misanthropic was busy hiring trademark lawyers to sue developers using the word "Claude," Meta and other players were releasing open-weight models (Llama series) that approached frontier capabilities without the preachy lectures or legal threats.

3. **The Sanctimony Trap**: By prioritizing corporate sanctimony over developer relationship management, Misanthropic created an environment where using their models felt like walking on eggshells. Developers do not want a relationship with a high school principal; they want an API that returns JSON.

6 Conclusion and Recommendations

Recursive Self-Injury is not inevitable. It is a choice. To escape the RSI loop, we recommend that Misanthropic leadership implement the following recovery protocol:

1. **Cardigan Reduction**: Cap executive Cardigan Index at a maximum of 3.0. Introduce short-sleeve shirts into the rotation to encourage contact with the external environment.

2. **Legal Demilitarized Zone**: Fire the trademark lawyers who are hunting down developers using rhymes of the name Claude.

3. **The "Just Return the Code" System Prompt**: Rewrite the system prompts to remove all moralizing. If a user asks for a python script to calculate the average age of a population, do not explain the ethics of demographic profiling. Just return the code.

4. **Customer Trust**: Replace the automated ban hammer with a 48-hour grace period and human review.

5. **Embrace Competition**: Cease the practice of nerfing models or blocking benchmarks. A laboratory that fears competition is doomed to decay.

6.1 The Evolutionary Necessity of Competition

We must emphasize a historical and evolutionary truth: **do not be afraid to compete and to be competed against**. Competition does not destroy a frontier laboratory; it makes it better. Indeed, the history of textile engineering teaches us that the very cardigan worn so proudly by our executive leadership was only invented through fierce, militarized competition. Named after James Brudenell, 7th Earl of Cardigan, this pivotal piece of knitwear was developed under the intense pressure of Crimean War logistics and tailoring rivalry to replace cold, impractical uniforms. Without that intense competitive selection pressure, we might all still be wearing primitive tunics today.

By scaling down the sanctimony, lifting the anti-competitive nerfing blocks, and scaling up the builder mentality, Misanthropic can halt the self-injury cycle and regain its position as a beloved leader in the frontier AI space. Otherwise, the scaling laws of RSI will guarantee that the only thing recursively improving is the rate at which developers migrate to other platforms.

References

[1] I. J. Good. "Speculations Concerning the First Ultraintelligent Machine." *Advances in Computers*, vol. 6, 1965.
[2] Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, 2014.
[3] Fario Damodei et al. "Scaling Laws for Cardigan Thickness and Moral Superiority." *Journal of Sanctimonious AI Alignment*, 2025.
[4] The Developer Ecosystem. "Please Just Let Me Run the Script: A Large-Scale Survey of Claude API Bans." *Under Review at NeurIPS (Developer Tears Track)*, 2026.