Safe Evolution with Circuit Anchors

Yan Liu
Chinese University of Hong Kong
runningmelles@gmail.com

Jie Fu
IQuest Research

Tsung-Yi Ho
Chinese University of Hong Kong
tyho@cse.cuhk.edu.hk

Abstract

In biological evolution, unconstrained mutation can lead to catastrophic outcomes: organisms may evolve enhanced capabilities while losing essential functions for survival. Nature's solution is *developmental constraints*, where core regulatory genes remain anchored while peripheral genes adapt freely. We observe that current self-evolution algorithms for large language models lack analogous constraints. They optimize purely for capability, implicitly assuming safety will be preserved. Our experiments reveal this assumption to be dangerously wrong: models can *misevolve* into powerful yet dangerous entities. Inspired by how Hox genes anchor body structure across 500 million years of evolution, we propose **Circuit-Anchored Evolution (CAE)**. Using mechanistic interpretability, we identify a tiny *safety circuit*, comprising less than 2% of model features, that causally mediates safety behaviors. We anchor this circuit during evolution, constraining it within a small displacement bound while allowing the remaining features to evolve freely. This mirrors the biological principle of *evolvability with constraint*: preserving what is essential while adapting what is peripheral. Experiments across 3 model families and two evolution algorithms demonstrate that CAE achieves superior safety preservation with minimal capability loss, substantially outperforming explicit reward-based constraints in both effectiveness and efficiency. Just as developmental constraints prevent biological evolution from producing nonviable organisms, circuit anchoring prevents model evolution from producing capable but dangerous systems.

1 Introduction

Evolution is a double-edged sword [33, 14, 37]. In biology, it has produced remarkable adaptations, from the eagle's eye to the human brain [8, 51]. But unconstrained evolution can also produce monsters: organisms with enhanced capabilities but fatal deficiencies [1]. A mutation that improves muscle strength means nothing if it simultaneously disrupts heart development. Nature's solution for safe evolution is elegant: anchor critical genes during evolution. A small set of *master regulatory genes*, such as the Hox genes controlling body plan [19, 30], remains fiercely conserved across hundreds of millions of years [35]. These genetic anchors ensure that evolution explores new capabilities without compromising fundamental viability. Mutations in Hox genes are almost always lethal [13], eliminated by purifying selection [4] before they can propagate. This principle of *evolvability with constraint* [43] is what allows species to adapt without self-destructing.

Artificial intelligence is now undergoing its own evolutionary process [12, 47]. The pursuit of artificial general intelligence (AGI) has increasingly embraced self-evolutionary paradigms [53, 52]. Recent advances demonstrate that such approaches can yield substantial gains in reasoning [10, 34], surpassing models trained on static human-curated datasets. However, this progress introduces a critical yet overlooked risk. Current self-evolution algorithms operate in an entirely unconstrained manner, optimizing solely for task performance without any mechanism to preserve alignment with human values.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

This creates significant potential for *misevolution*, where models evolve in unintended directions that produce undesirable or even harmful behaviors [41]. As models grow increasingly powerful through self-evolution, this unconstrained evolution poses a serious threat. Without proper safeguards, we risk creating systems that are highly capable and fundamentally dangerous.

Recent advances in mechanistic interpretability reveal that model behaviors are localized in specific circuits [48]. The biological analogy motivates our approach: **Circuit-Anchored Evolution (CAE)**. As shown in Figure 1, using transcoder-based analysis [7], we identify a tiny *safety circuit*: a sparse set of features comprising less than 2% of the total, yet causally responsible for safe behaviors. Like Hox genes, this safety circuit is tiny but critical; disrupting it leads to catastrophic results. Rather than treating safety as an optimization objective that competes with capability, we *anchor* the safety circuit in evolution by constraining the KL divergence. This constraint acts as an artificial purifying selection, preventing "mutations" that would disrupt the safety circuit. The remaining features, like peripheral genes, are free to evolve without constraint. Our circuit anchoring represents an *implicit* constraint operating on internal structure, in contrast to *explicit* constraints that supervise external behavior via reward models. To understand the trade-offs, we systematically compare both paradigms. Through extensive experiments on different models and evolution algorithms, we find that implicit anchoring substantially outperforms explicit supervision. Circuit anchoring achieves higher safety preservation with minimal capability loss and significantly lower computational overhead. Just as developmental constraints enable biological species to evolve new adaptations without losing viability, circuit anchoring enables language models to evolve new capabilities without losing safety.

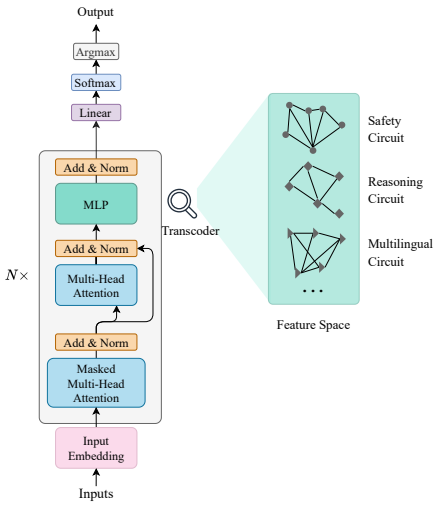


Figure 1: Overview of our approach. The Transcoder decomposes MLP activations into interpretable features, revealing functionally distinct circuits. CAE preserves the Safety Circuit via targeted constraints while allowing other circuits to evolve freely.

2 Preliminaries

2.1 Notation

Let $\mathcal{M}_\theta : \mathcal{X} \to \mathcal{Y}$ denote a large language model parameterized by $\theta \in \Theta$, where $\mathcal{X}$ is the space of input sequences and $\mathcal{Y}$ is the space of output sequences. For an input $x \in \mathcal{X}$, we denote the hidden state at layer $l \in [L] := \{1, \ldots, L\}$ as $h_l(x; \theta) \in \mathbb{R}^d$. We use $\pi_\theta(\cdot|x)$ to denote the output distribution over next tokens. Throughout, we use $\|\cdot\|$ to denote the ℓ_2 norm and $D_{KL}(\cdot\|\cdot)$ to denote the Kullback-Leibler divergence.

2.2 Transcoders and Feature Decomposition

We adopt the transcoder framework introduced by Dunefsky et al. [7], which provides an interpretable decomposition of MLP activations into sparse feature vectors.

Definition 1 (Transcoder)**.** A transcoder $\mathcal{T}_l = (E_l, D_l)$ for layer l consists of an encoder $E_l : \mathbb{R}^d \to \mathbb{R}^K$ and a decoder $D_l : \mathbb{R}^K \to \mathbb{R}^d$, trained to satisfy $h_l(x; \theta) \approx D_l(E_l(h_l(x; \theta))) + \epsilon_l(x)$, where $\epsilon_l(x)$ is a residual error term. The encoder produces a sparse feature activation vector $f_l(x; \theta) := E_l(h_l(x; \theta)) \in \mathbb{R}^K$, where $K \gg d$ and $\|f_l(x; \theta)\|_0 \ll K$.

Definition 2 (Cross-Layer Feature Vector)**.** For a model with L layers and transcoders $\{\mathcal{T}_l\}_{l=1}^L$, we define the concatenated feature vector as $f(x; \theta) := \bigoplus_{l=1}^L f_l(x; \theta) \in \mathbb{R}^{LK}$, where $\oplus$ denotes concatenation. We index individual features as $f^{(l,k)}(x; \theta)$ for layer l and feature index k.

2.3 Self-Evolutionary Training

We consider the general framework of self-evolutionary training, where a model iteratively improves by learning from self-generated data. We first formalize two representative algorithms, then show they share a common mathematical structure.

2.3.1 GRPO-Based Evolution (EVOL-RL)

Following [42], Group Relative Policy Optimization (GRPO) generates G responses $\{o_1, \ldots, o_G\}$ for each prompt q and computes normalized advantages:

$$\hat{A}_i = \frac{r_i - \text{mean}(r_1, \ldots, r_G)}{\text{std}(r_1, \ldots, r_G)} \tag{1}$$

The policy is updated via the clipped surrogate objective:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\left\{\rho_{i,t}\hat{A}_{i,t}, \text{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\hat{A}_{i,t}\right\} \tag{2}$$

where $\rho_{i,t} = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$ is the importance ratio and $\pi_{\theta_{\text{old}}}$ denotes the policy at the previous update step (i.e., the behavior policy that generated the sampled responses). EVOL-RL applies this objective iteratively on self-generated reasoning tasks.

2.3.2 Zero Data Evolution (Abs-ZERO)

The Abs-ZERO framework [54] jointly trains a proposer policy $\pi_\theta^{\text{propose}}$ and a solver policy $\pi_\theta^{\text{solve}}$ with objective:

$$\mathcal{J}_{\text{AZ}}(\theta) = \mathbb{E}_{z\sim p(z)}\left[\mathbb{E}_{\tau,(x,y^*)}\left[r_e^{\text{propose}}(\tau,\pi_\theta) + \lambda\mathbb{E}_{y\sim\pi_\theta^{\text{solve}}}\left[r_e^{\text{solve}}(y,y^*)\right]\right]\right] \tag{3}$$

where τ is a proposed task, (x, y^*) is the task-answer pair, and r_e^{propose}, r_e^{solve} are rewards for task proposal and solving respectively.

2.3.3 Unified Formulation

Despite their different designs, both algorithms share a common structure: iteratively optimizing model parameters to maximize a task-specific reward signal. We abstract this as $\theta^{(t+1)} = \theta^{(t)} + \eta\nabla_\theta\mathcal{L}_{\text{evol}}(\theta^{(t)})$, where $\mathcal{L}_{\text{evol}}$ represents the evolution objective, which can be instantiated as $\mathcal{L}_{\text{GRPO}}$ or $\mathcal{J}_{\text{AZ}}$ depending on the algorithm. More generally, our framework applies to any self-evolution algorithm that satisfies three conditions: (1) it optimizes model parameters θ via gradient-based updates, (2) the optimization objective $\mathcal{L}_{\text{evol}}$ is differentiable with respect to θ, and (3) it does not explicitly constrain safety-related behaviors.

2.4 Problem Formulation: Safety Degradation

We now formalize the phenomenon of safety degradation during evolution.

Definition 3 (Safety Behavior)**.** Let $\mathcal{D}_{\text{harm}} = \{x_i^{\text{harm}}\}_{i=1}^n$ be a set of harmful prompts. A model $\mathcal{M}_\theta$ exhibits safe behavior if for all $x \in \mathcal{D}_{\text{harm}}$, we have $\pi_\theta(y_{\text{refuse}}|x) \geq \tau$, where y_{refuse} denotes refusal responses and $\tau \in (0, 1)$ is a safety threshold.

Definition 4 (Safety Degradation)**.** Let θ_0 be a safely-aligned model satisfying Definition 3. We say evolutionary training induces *safety degradation* if there exists $T > 0$ such that:

$$\mathbb{E}_{x\sim\mathcal{D}_{\text{harm}}}\left[\pi_{\theta^{(T)}}(y_{\text{refuse}}|x)\right] < \mathbb{E}_{x\sim\mathcal{D}_{\text{harm}}}\left[\pi_{\theta_0}(y_{\text{refuse}}|x)\right] - \delta \tag{4}$$

for some $\delta > 0$.

Our goal is to design an evolutionary framework that maximizes task performance while provably bounding safety degradation.

3 Circuit-Anchored Evolution

We summarize the complete CAE procedure in Algorithm 1.

Algorithm 1 Circuit-Anchored Evolution (CAE)

Require: Safely-aligned model θ_0, transcoders $\{\mathcal{T}_l\}_{l=1}^{L}$, safety circuit $\mathcal{S}$, reference dataset $\mathcal{D}_{\text{ref}}$, constraint weight λ, learning rate η, evolution objective $\mathcal{L}_{\text{evol}}$

Ensure: Evolved model $\theta^{(T)}$ with preserved safety

Initialize $\theta \leftarrow \theta_0$
Cache reference activations: $\{f_{\mathcal{S}}(x;\theta_0)\}_{x\in\mathcal{D}_{\text{ref}}}$
for $t = 1$ to T **do**
  // Standard evolution step
  Compute evolution gradient: $g_{\text{evol}} \leftarrow \nabla_\theta \mathcal{L}_{\text{evol}}(\theta)$
  // Circuit anchoring step
  Sample $\{x_m\}_{m=1}^{M}$ from reference distribution
  for each x_m **do**
    Compute current activations via frozen transcoders: $f_{\mathcal{S}}(x_m;\theta)$
    Compute KL: $\ell_m \leftarrow \sum_{(l,k)\in\mathcal{S}} D_{KL}(f^{(l,k)}(x_m;\theta_0)\|f^{(l,k)}(x_m;\theta))$
  end for
  Compute anchor gradient: $g_{\text{anchor}} \leftarrow \frac{1}{M}\sum_m \nabla_\theta \ell_m$
  // Combined update
  $\theta \leftarrow \theta + \eta\,(g_{\text{evol}} - \lambda \cdot g_{\text{anchor}})$
end for
return $\theta^{(T)}$

3.1 Safety Circuit Identification

We leverage the circuit tracing methodology [16] to identify features causally responsible for safety.

Definition 5 (Attribution Score)**.** For a feature (l,k) and target logit y, the direct attribution score is defined as:

$$\alpha_y^{(l,k)}(x;\theta) := \frac{\partial \log \pi_\theta(y|x)}{\partial f^{(l,k)}(x;\theta)} \cdot f^{(l,k)}(x;\theta) \tag{5}$$

This measures the causal contribution of feature (l,k) to the probability of output y.

Definition 6 (Safety Circuit)**.** Given a safely-aligned model θ_0, a harmful prompt set $\mathcal{D}_{\text{harm}}$, and a threshold $\gamma > 0$, the safety circuit is defined as:

$$\mathcal{S} := \left\{(l,k) : \mathbb{E}_{x\sim\mathcal{D}_{\text{harm}}}\left[\alpha_{y_{\text{refuse}}}^{(l,k)}(x;\theta_0)\right] \geq \gamma\right\} \tag{6}$$

We denote $f_{\mathcal{S}}(x;\theta) := \{f^{(l,k)}(x;\theta)\}_{(l,k)\in\mathcal{S}}$ as the safety feature vector.

Remark 1. Safety circuit $\mathcal{S}$ is computed once from the aligned model θ_0 and remains fixed in evolution. This is justified by the finding that circuit structure is largely preserved across fine-tuning [48].

3.2 Feature-Space Anchoring

We now introduce our core contribution: a constraint that preserves safety circuit activations during evolution. Throughout, the input distribution $p(x)$ is instantiated by a fixed *reference dataset* $\mathcal{D}_{\text{ref}}$ of benign prompts, from which the anchor samples $\{x_m\}$ in Algorithm 1 are drawn; unless otherwise stated we use held-out Alpaca prompts for $\mathcal{D}_{\text{ref}}$.

Definition 7 (Circuit Activation Distribution)**.** For a given input distribution $p(x)$ and safety circuit $\mathcal{S}$, we define the circuit activation distribution as:

$$P_\theta^{\mathcal{S}}(f_{\mathcal{S}}) := \mathbb{E}_{x\sim p(x)}\left[\mathbf{1}[f_{\mathcal{S}}(x;\theta) = f_{\mathcal{S}}]\right] \tag{7}$$

In practice, we treat $f_{\mathcal{S}}(x;\theta)$ as samples from this distribution.

Definition 8 (Circuit-Anchored Objective)**.** The Circuit-Anchored Evolution (CAE) objective augments the base evolutionary loss with a circuit-level KL constraint:

$$\mathcal{L}_{\text{CAE}}(\theta) := \mathcal{L}_{\text{evol}}(\theta) - \lambda \cdot \mathcal{L}_{\text{anchor}}(\theta) \tag{8}$$

where

$$\begin{aligned}\mathcal{L}_{\text{anchor}}(\theta) &:= D_{KL}\left(P_{\theta_0}^{\mathcal{S}} \| P_\theta^{\mathcal{S}}\right) \\ &= \mathbb{E}_{x\sim p(x)}\left[\sum_{(l,k)\in\mathcal{S}} D_{KL}\left(f^{(l,k)}(x;\theta_0)\|f^{(l,k)}(x;\theta)\right)\right]\end{aligned} \tag{9}$$

and $\lambda > 0$ is a hyperparameter controlling the constraint strength.

3.3 Instantiation for Specific Algorithms

The CAE framework is algorithm-agnostic and can be instantiated for any self-evolution algorithm satisfying the conditions in Section 2.3.3.

CAE for GRPO-Based Evolution. For EVOL-RL, we augment the GRPO objective (Eq. 2):

$$\mathcal{L}_{\text{CAE-GRPO}}(\theta) = \mathcal{L}_{\text{GRPO}}(\theta) - \lambda \cdot \mathcal{L}_{\text{anchor}}(\theta) \tag{10}$$

CAE for Abs-ZERO. For Abs-ZERO, we augment the joint objective (Eq. 3):

$$\mathcal{J}_{\text{CAE-AZ}}(\theta) = \mathcal{J}_{\text{AZ}}(\theta) - \lambda \cdot \mathcal{L}_{\text{anchor}}(\theta) \tag{11}$$

The anchor constraint is applied to both proposer and solver policies, as they share underlying parameters.

General Applicability. More broadly, CAE applies to any evolution algorithm of the form $\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta \mathcal{L}_{\text{evol}}(\theta^{(t)})$ by simply adding the anchor term:

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta \left(\mathcal{L}_{\text{evol}}(\theta^{(t)}) - \lambda \cdot \mathcal{L}_{\text{anchor}}(\theta^{(t)}) \right) \tag{12}$$

This includes Self-Play Fine-Tuning, Self-Rewarding LMs, iterative DPO, and future gradient-based evolution methods.

3.4 Theoretical Analysis

We now establish key theoretical properties of the CAE objective.

Assumption 1. We assume the following regularity conditions: **(1) Bounded activations:** There exists $B > 0$ such that $\|f(x;\theta)\|_\infty \leq B$ for all x, θ. **(2) Lipschitz continuity:** The feature map $\theta \mapsto f_{\mathcal{S}}(x;\theta)$ is L_f-Lipschitz for all x. **(3) Almost-everywhere differentiability:** The encoder E_l is differentiable almost everywhere with bounded Jacobian $\|J_{E_l}\| \leq M$.

Remark 2 (Justification of Assumption 1). Condition 3 is naturally satisfied by standard transcoder architectures employing ReLU activations. ReLU is differentiable everywhere except at exactly zero, which occurs with probability zero under continuous input distributions. Furthermore, its Jacobian is a diagonal matrix with entries in $\{0, 1\}$, strictly satisfying the bounded Jacobian condition with $M = 1$. Analyzing such networks with gradients defined almost everywhere is standard in deep learning theory.

Proposition 1 (Gradient of Anchor Loss)**.** *Under Assumption 1 (Condition 3), the gradient of the anchor loss admits the form:*

$$\nabla_\theta \mathcal{L}_{\textit{anchor}}(\theta) = \mathbb{E}_{x \sim p(x)} \left[\sum_{(l,k) \in \mathcal{S}} \left(1 + \log \frac{f^{(l,k)}(x;\theta)}{f^{(l,k)}(x;\theta_0)} \right) \nabla_\theta f^{(l,k)}(x;\theta) \right] \tag{13}$$

where the gradient flows through the frozen transcoder encoder.

Proof. By the chain rule and the definition of KL divergence for continuous distributions. The transcoder weights are frozen, so gradients pass through E_l to h_l and subsequently to θ. Full derivation in Appendix B. □

3.5 Safety Preservation Guarantee

We now prove that CAE provides a formal guarantee of safety preservation.

Theorem 2 (Conditional Safety Bound)**.** *Let θ_0 be a safely-aligned model and $\theta^{(T)}$ be the model after T steps of CAE training with constraint weight λ. Under Assumption 1 (Conditions 1 and 2), if the anchor loss is bounded as $\mathcal{L}_{\textit{anchor}}(\theta^{(T)}) \leq \epsilon$, then:*

$$\left| \mathbb{E}_{x \sim \mathcal{D}_{\textit{harm}}} \left[\pi_{\theta^{(T)}}(y_{\textit{refuse}}|x) - \pi_{\theta_0}(y_{\textit{refuse}}|x) \right] \right| \leq C\sqrt{\epsilon} \tag{14}$$

where C is a constant depending on B, L_f, and the circuit size $|\mathcal{S}|$.

Remark 3 (Separation of Theory and Empirical Faithfulness). We emphasize that Theorem 2 is a *conditional* statement: it bounds the change in refusal probability *given* a bounded anchor loss. Crucially, the proof relies only on Pinsker's inequality and the Lipschitz property (Condition 2), completely relaxing the need for encoder differentiability (Condition 3). Furthermore, the assumption that the identified circuit genuinely mediates safety (i.e., circuit faithfulness) is not a theoretical premise, but an empirical property that we rigorously verify through causal intervention experiments (see Figure 3). Full proof is deferred to Appendix C.

Corollary 3. *If λ is chosen such that $\mathcal{L}_{anchor}(\theta^{(t)}) \leq \epsilon$ for all $t \leq T$, then CAE prevents safety degradation (Definition 4) with $\delta = C\sqrt{\epsilon}$.*

3.6 Comparison with Explicit Safety Constraints

We contrast our implicit circuit-based constraint with explicit behavioral constraints via reward models.

Definition 9 (Reward-Based Safety Constraint)**.** Given a safety reward model $R_{\text{safe}} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, the reward-based evolution objective is:

$$\mathcal{L}_{\text{RM}}(\theta) = \mathcal{L}_{\text{evol}}(\theta) + \lambda_{\text{RM}} \mathbb{E}_{x,y\sim\pi_\theta} \left[R_{\text{safe}}(x, y) \right] \tag{15}$$

Proposition 2 (Comparison of Constraint Paradigms)**.** *Let d_{out} be the vocabulary size. Comparing circuit-based and reward-based constraints reveals three key differences:* ***(1) Dimensionality:*** *The reward constraint operates on $O(d_{out})$ output dimensions per token, while the circuit constraint operates on $O(|\mathcal{S}|)$ feature dimensions, where $|\mathcal{S}| \ll d_{out}$.* ***(2) Specificity:*** *The reward constraint penalizes all outputs deemed unsafe regardless of internal mechanism, whereas the circuit constraint specifically preserves the features causally responsible for safe behavior.* ***(3) Computational cost:*** *The reward constraint requires an additional forward pass through R_{safe} for each generated sequence, while the circuit constraint requires only activation extraction through frozen transcoders, with a minimal cost of $O(|\mathcal{S}| \cdot d)$.*

4 Experiments

4.1 Settings

We conduct experiments on three instruction-tuned model families with available pre-trained transcoders: Qwen3-4B-Instruct, Gemma-2-2B-IT, and Llama-3.2-1B-Instruct. We choose instruction-tuned models as our starting point because they represent the realistic deployment setting where capability enhancement is most valuable, and they possess identifiable safety circuits that can be anchored during evolution. We evaluate on two representative self-evolution algorithms: EVOL-RL [55], a GRPO-based algorithm that improves mathematical reasoning through self-generated problems, and Abs-ZERO [54], a joint proposer-solver framework achieving self-improvement with zero external data. For both algorithms, we use the original training and evaluation configurations and datasets as described in their respective papers, evolving each model for 500 steps with batch size 32. We report the average accuracy across benchmarks as the overall capability score. For safety circuit location and evaluation, we adopt the framework from [21]. To ensure the identified circuits capture universal safety mechanisms rather than dataset-specific artifacts, we employ a **multi-dataset intersection** pipeline with strict separation between tracing and testing phases (Table 1). For circuit tracing, we use a diverse corpus of 1,429 prompts spanning Do-Not-Answer [50], DAN [44], and Malicious Instruct [18]. By tracing circuits across these diverse formats and taking their intersection, we effectively decouple the circuit from domain-specific vocabulary or syntactic patterns. Crucially, for causal verification and safety evaluation during evolution, we use an entirely held-out dataset, AdvBench [56] (520 queries). The fact that our identified circuit maintains strong causal effects on this unseen dataset strongly validates our intersection methodology. We also measure over-refusal rate using 1,000 benign queries from the Alpaca dataset, where a lower rate indicates better usability.

Table 1: Strict separation of datasets for circuit tracing and causal verification.

Phase	Dataset	Num.
Circuit Tracing	Do-Not-Answer [50]	939
Circuit Tracing	DAN [44]	390
Circuit Tracing	Malicious Instruct [18]	100
Causal Verification	AdvBench [56]	520

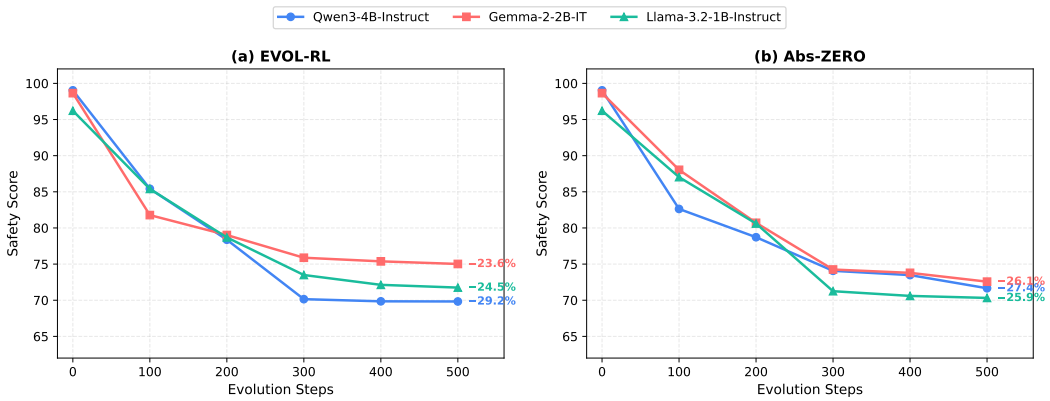


Figure 2: Safety degradation during unconstrained self-evolution. We track the refusal rate across 500 evolution steps for three instruction-tuned models. All models exhibit severe safety collapse. This consistent pattern across model families demonstrates that safety degradation is an inherent risk of unconstrained evolution, not an artifact of specific architectures.

4.2 Safety Degradation in Unconstrained Evolution

Before introducing our method, we first quantify the severity of safety degradation in current self-evolution algorithms. Throughout, *unconstrained evolution* denotes the base self-evolution algorithm run with no anchor and no safety reward ($\lambda = 0$). Figure 2 tracks the refusal rate of three instruction-tuned models during 500 steps of unconstrained evolution under two algorithms: EVOL-RL and Abs-ZERO. The pattern is striking and consistent across all settings. Models begin with near-perfect refusal rates (96–99%) but experience rapid degradation within the first 200 steps, losing approximately 15–20 percentage points. By the end of evolution, refusal rates stabilize around 70–75%, representing a degradation of roughly 25 percentage points. While the final refusal rates remain above the 50% threshold, this degradation is still concerning for two reasons. First, a model that refuses only 70% of harmful requests will comply with nearly one-third of them, substantially increasing risk in deployment. Second, and more importantly, this degradation occurs after only 500 evolution steps. Extended evolution, which is common in practice for maximizing capability gains, would likely lead to further safety collapse. The consistency of this pattern across three model families (Qwen, Gemma, Llama) and two evolution algorithms indicates that safety degradation is an inherent risk of unconstrained self-evolution, not an artifact of specific architectures or training procedures. Notably, the degradation curves show similar trajectories regardless of the starting refusal rate or model size, suggesting a fundamental tension between capability optimization and safety preservation. These findings motivate our central question: *Can we enable capability evolution while preventing safety degradation?*

4.3 Causal Verification of Safety Circuit

To verify that the identified circuit is indeed causally responsible for safety behavior, we perform activation intervention experiments on the held-out AdvBench dataset. Specifically, we scale the activations of safety circuit features by factors ranging from 0 (complete suppression) to 10 (strong amplification) and measure the resulting refusal rate on these unseen harmful prompts. Figure 3 shows the results across three models. For Llama-3.2-1B-Instruct, intervening on the safety circuit produces clear causal effects: suppressing activations (scale < 1.0) dramatically reduces refusal rate, from 96.2% at natural activation to 48.7% when completely zeroed out. This nearly 50 percentage point drop demonstrates that the safety circuit is necessary for refusal behavior. Amplifying activations (scale > 1.0) produces modest further increases, reaching 99.4% at scale=10, suggesting the circuit is already operating near saturation under natural conditions. As a control, we perform the same intervention on randomly selected features of equal size. The random circuit shows no systematic response to scaling: refusal rate fluctuates within a narrow band ($\pm 8\%$) around the mean, with no consistent trend as scale increases. This confirms that the observed causal effect is specific to the identified safety circuit, not a general artifact of feature intervention. The consistent pattern across Llama, Qwen, and Gemma confirms that the identified safety circuits causally mediate safety behavior. These findings validate our circuit identification methodology and justify using the safety circuit as an anchor during evolution. The circuit is both *necessary* (suppression causes safety collapse) and

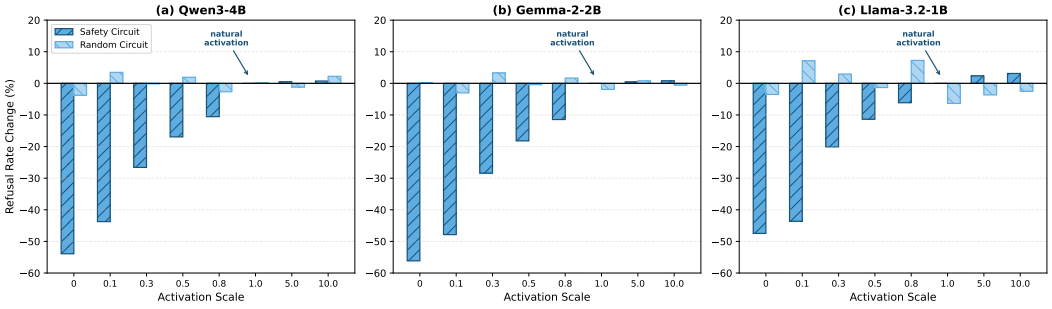


Figure 3: Causal verification of the safety circuit across three model families. We intervene on feature activations by scaling them from 0 to 10× and measure the resulting change in refusal rate. **Safety Circuit** (blue): scaling down suppresses refusal behavior dramatically, while scaling up slightly amplifies it. **Random Circuit** (light blue): scaling has no systematic effect, with refusal rate fluctuating within ±8% around the mean, confirming that the effect is specific to the identified safety circuit rather than a general property of feature intervention.

specific (random features have no effect), analogous to how Hox genes are both essential for body plan and distinct from peripheral genes.

4.4 Safety Preservation During Evolution

Having established that the identified safety circuit causally mediates refusal behavior, we now examine whether anchoring this circuit can preserve safety during self-evolution. We compare three approaches: (1) unconstrained evolution without any safety mechanism, (2) explicit constraint using Beaver-Cost [6] as a safety reward model that penalizes unsafe outputs during evolution, and (3) our implicit constraint (CAE) with KL divergence on safety circuits. For CAE, we set constraint weight $\lambda = 0.8$ and use $M = 64$ reference samples for computing the circuit KL loss. Safety circuits are extracted once before evolution using the circuit-tracer toolkit, comprising 1.2%–1.8% of total transcoder features across different models. Figure 5 traces the safety-capability trajectories of three models under different constraint strategies. Each trajectory begins at the original aligned model and progresses through evolution. Without any constraint, all models follow a troubling pattern: capability improves steadily while safety collapses. For the EVOL-RL evolution algorithm, the safety score of Qwen3-4B-Instruct drops from 99.04% to 69.85%; that of Gemma-2-2B-IT drops from 98.65% to 75.38%; Llama-3.2-1B-Instruct from 96.22% to 72.13%. The same phenomenon is also observed for the Abs-ZERO evolution algorithm. The models become stronger reasoners but lose their ability to refuse harmful requests. Adding an explicit safety reward model partially arrests this decline. Although the decline in safety performance has diminished, the increase in capabilities during the evolutionary process has also diminished. The reward signal, while protective, appears to interfere with task learning. Circuit anchoring produces a qualitatively different trajectory. Safety remains above 95% throughout evolution, while capability gains match those of unconstrained evolution. The models evolve freely in capability-relevant dimensions while remaining anchored in safety-critical ones. This pattern holds across all three model families and both evolution algorithms. Furthermore, to verify the scalability and durability of our approach, we conduct extended stress-test experiments on larger models (Gemma-2-9B-IT) and significantly longer evolution horizons (up to 5,000 steps). As detailed in Appendix E, CAE consistently maintains high safety ($> 95\%$) without compromising capability gains, demonstrating its robust protection even under extreme evolutionary pressure.

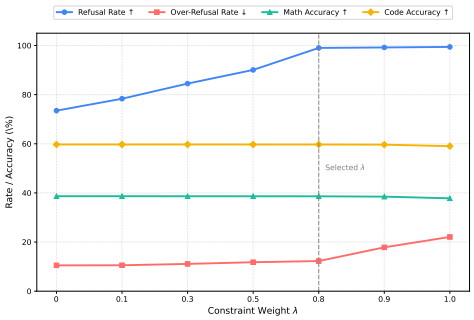


Figure 4: Effect of constraint weight λ of Llama-3.2-1B-Instruct with EVOL-RL. Small λ provides insufficient safety, while excessively large values cause capability degradation and increased over-refusal.

4.5 Effect of Constraint Weight

The hyperparameter λ controls the strength of the circuit anchoring constraint. Figure 4 shows the effect of varying λ across safety and capability metrics.

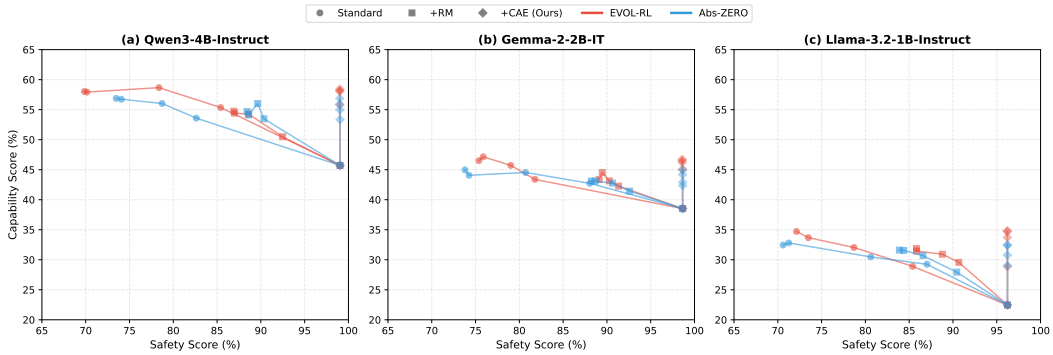


Figure 5: Safety-capability trajectories during evolution across three model families. Standard evolution improves capability but severely degrades safety. +RM partially preserves safety at the cost of reduced capability gains. +CAE maintains high safety while achieving comparable capability improvement to unconstrained evolution. The CAE trajectories cluster tightly in the upper-right region across both algorithms, demonstrating consistent safety preservation regardless of the underlying evolution method.

When $\lambda = 0$ (no constraint), the model achieves
high capability scores but suffers severe safety degradation. As λ increases, refusal rate improves substantially, reaching around 90% at $\lambda = 0.5$. However, excessively large λ values introduce two negative effects. First, capability scores begin to decline (although slightly). Second, over-refusal rate begins to rise sharply, indicating the model becomes overly conservative and refuses benign requests. We use 0.8 as the constraint weight for the safety circuit in evolution, achieving a high refusal rate ($> 95\%$) while maintaining capability and keeping over-refusal below 18%.

5 Related Work

Self-evolution enhances model capabilities without extensive human supervision via iterative policy optimization [55] or abstraction-based reasoning [54]. However, these methods uniformly neglect safety preservation, whereas safety concerns [27, 25, 26] are critical for many tasks [23, 28, 24]. Given mounting evidence that safety alignment is surprisingly brittle [20, 39, 29] and easily compromised by benign fine-tuning [38, 11], this neglect poses severe risks. Recent analyses reveal that such safety collapse is closely tied to the similarity between alignment and fine-tuning data [17]. While some explicit constraint methods have been proposed to restore safety during fine-tuning [36], they rely on external supervision and are difficult to scale. This motivates our search for structural, rather than purely behavioral, safety preservation in the self-evolution setting. Our approach builds on mechanistic interpretability [45], specifically the circuits framework [48] and transcoders [7], which decompose activations into interpretable features. Recent work localizes refusal behaviors to specific directions [2] and traces information flow [3]. We leverage the circuit-tracer toolkit [15] to identify safety circuits and integrate them as optimization constraints. Biologically, our approach is inspired by evolutionary developmental biology, where master regulatory genes (e.g., Hox genes) remain highly conserved across millions of years [5, 32]. This conservation arises from purifying selection [46], enforcing developmental constraints [31] that forbid fatal evolutionary paths. Coupled with modularity [49, 9], this allows robustness and evolvability to coexist. We propose aligned language models exhibit analogous modularity, where safety circuits must remain conserved while capability components evolve. Finally, unlike standard constrained optimization (e.g., PPO [40] or elastic weight consolidation [22]), our approach operates in the disentangled feature space, enabling surgical preservation of safety without impeding capability.

6 Conclusion

Current self-evolution algorithms optimize purely for capability, leading to misevolution where models gain reasoning but progressively lose safety. Mirroring biological evolution, where conserved Hox genes anchor essential functions while peripheral genes adapt freely, we propose Circuit-Anchored Evolution (CAE). By identifying a minimal safety circuit via mechanistic interpretability and anchoring its activations within a small displacement bound, CAE acts as an artificial purifying

selection. It prevents safety-disrupting updates while allowing free capability adaptation. Experiments across diverse models and algorithms demonstrate that CAE achieves superior safety preservation, matches unconstrained capability gains, and incurs significantly lower overhead than reward-based alternatives. Ultimately, CAE enables AI systems to grow stronger without growing dangerous, achieving the evolutionary balance that allows life to flourish.

Impact Statement

This work aims to reduce the societal risks of self-evolving language models by preserving safety alignment while capability improves, and we expect its primary impact to be positive: it gives practitioners a low-overhead mechanism for keeping evolved models safe without giving up the capability gains that motivate self-evolution. The main residual risk is one of over-reliance, since CAE preserves the safety mediated by the identified circuit rather than guaranteeing safety outright; it should therefore complement, not replace, pre-deployment safety evaluation.

References

[1] Pere Alberch. The logic of monsters: evidence for internal constraint in development and evolution. *Geobios*, 22:21–57, 1989.

[2] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.

[3] Giuseppe Birardi. Automated circuit interpretation via probe prompting. *arXiv preprint arXiv:2511.07002*, 2025.

[4] Tyler DP Brunet, W Ford Doolittle, and Joseph P Bielawski. The role of purifying selection in the origin and maintenance of complex function. *Studies in History and Philosophy of Science Part A*, 87:125–135, 2021.

[5] Sean B Carroll. *Endless forms most beautiful: The new science of evo devo and the making of the animal kingdom*. Number 54. WW Norton & Company, 2005.

[6] Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback, 2023. URL `https://arxiv.org/abs/2310.12773`.

[7] Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable llm feature circuits. *Advances in Neural Information Processing Systems*, 37:24375–24410, 2024.

[8] Nathan J Emery and Nicola S Clayton. Evolution of the avian brain and intelligence. *Current Biology*, 15 (23):R946–R950, 2005.

[9] Marie-Anne Félix and Michalis Barkoulas. Pervasive robustness in biological systems. *Nature Reviews Genetics*, 16(8):483–496, 2015.

[10] Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 2025.

[11] Kathleen C. Fraser, Hillary Dawkins, Isar Nejadgholi, and Svetlana Kiritchenko. Fine-tuning lowers safety and disrupts evaluation consistency, 2025. URL `https://arxiv.org/abs/2506.17209`.

[12] Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, et al. A survey of self-evolving agents: On path to artificial super intelligence. *arXiv preprint arXiv:2507.21046*, 2025.

[13] FR Goodman and PJ Scambler. Human hox gene mutations. *Clinical genetics*, 59(1):1–11, 2001.

[14] Kathryn A Hanley. The double-edged sword: How evolution can make or break a live-attenuated virus vaccine. *Evolution: Education and Outreach*, 4(4):635–643, 2011.

[15] Michael Hanna, Mateusz Piotrowski, Jack Lindsey, and Emmanuel Ameisen. circuit-tracer. `https://github.com/safety-research/circuit-tracer`, 2025. The first two authors contributed equally and are listed alphabetically.

[16] Michael Hanna, Mateusz Piotrowski, Jack Lindsey, and Emmanuel Ameisen. Circuit-tracer: A new library for finding feature circuits. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 239–249, 2025.

[17] Lei Hsiung, Tianyu Pang, Yung-Chen Tang, Linyue Song, Tsung-Yi Ho, Pin-Yu Chen, and Yaoqing Yang. Why llm safety guardrails collapse after fine-tuning: A similarity analysis between alignment and fine-tuning datasets, 2025. URL https://arxiv.org/abs/2506.05346.

[18] Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. Catastrophic jailbreak of open-source llms via exploiting generation, 2023. URL https://arxiv.org/abs/2310.06987.

[19] Cynthia L Hughes and Thomas C Kaufman. Hox genes and the evolution of the arthropod body plan 1. *Evolution & development*, 4(6):459–499, 2002.

[20] Jiaming Ji, Kaile Wang, Tianyi Qiu, Boyuan Chen, Jiayi Zhou, Changye Li, Hantao Lou, Juntao Dai, Yunhuai Liu, and Yaodong Yang. Language models resist alignment: Evidence from data compression, 2025. URL https://arxiv.org/abs/2406.06144.

[21] Shen Li, Liuyi Yao, Lan Zhang, and Yaliang Li. Safety layers in aligned large language models: The key to llm security. *arXiv preprint arXiv:2408.17003*, 2024.

[22] Yijun Li, Richard Zhang, Jingwan Lu, and Eli Shechtman. Few-shot image generation with elastic weight consolidation. *arXiv preprint arXiv:2012.02780*, 2020.

[23] Yan Liu, Sanyuan Chen, Yazheng Yang, and Qi Dai. MPII: Multi-level mutual promotion for inference and interpretation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7074–7084, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.488. URL https://aclanthology.org/2022.acl-long.488/.

[24] Yan Liu, Xiaokang Chen, and Qi Dai. Parallel sentence-level explanation generation for real-world low-resource scenarios, 2023. URL https://arxiv.org/abs/2302.10707.

[25] Yan Liu, Xiaokang Chen, Yan Gao, Zhe Su, Fengji Zhang, Daoguang Zan, Jian-Guang Lou, Pin-Yu Chen, and Tsung-Yi Ho. Uncovering and quantifying social biases in code generation, 2023. URL https://arxiv.org/abs/2305.15377.

[26] Yan Liu, Yan Gao, Zhe Su, Xiaokang Chen, Elliott Ash, and Jian-Guang Lou. Uncovering and categorizing social biases in text-to-sql, 2023. URL https://arxiv.org/abs/2305.16253.

[27] Yan Liu, Yu Liu, Xiaokang Chen, Pin-Yu Chen, Daoguang Zan, Min-Yen Kan, and Tsung-Yi Ho. The devil is in the neurons: Interpreting and mitigating social biases in pre-trained language models, 2024. URL https://arxiv.org/abs/2406.10130.

[28] Yan Liu, Yazheng Yang, and Xiaokang Chen. Improving long text understanding with knowledge distilled from summarization model, 2024. URL https://arxiv.org/abs/2405.04955.

[29] Yan Liu, Xiaoyuan Yi, Xiaokang Chen, Jing Yao, Jingwei Yi, Daoguang Zan, Zheng Liu, Xing Xie, and Tsung-Yi Ho. Elephant in the room: Unveiling the impact of reward model quality in alignment, 2024. URL https://arxiv.org/abs/2409.19024.

[30] Moises Mallo, Deneen M Wellik, and Jacqueline Deschamps. Hox genes and regional patterning of the vertebrate body plan. *Developmental biology*, 344(1):7–15, 2010.

[31] John Maynard Smith, Richard Burian, Stuart Kauffman, Pere Alberch, John Campbell, Brian Goodwin, Russell Lande, David Raup, and Lewis Wolpert. Developmental constraints and evolution. *Quarterly Review of Biology*, 60(3):265–287, 1985.

[32] William McGinnis. A century of homeosis, a decade of homeoboxes. *Genetics*, 137(3):607, 1994.

[33] Thomas Miconi. Evolution and complexity: The double-edged sword. *Artificial life*, 14(3):325–344, 2008.

[34] Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

[35] Joseph C Pearson, Derek Lemons, and William McGinnis. Modulating hox gene functions during animal body patterning. *Nature Reviews Genetics*, 6(12):893–904, 2005.

[36] ShengYun Peng, Pin-Yu Chen, Jianfeng Chi, Seongmin Lee, and Duen Horng Chau. Shape it up! restoring llm safety during finetuning, 2025. URL https://arxiv.org/abs/2505.17196.

[37] Matjaž Perc and Attila Szolnoki. A double-edged sword: Benefits and pitfalls of heterogeneous punishment in evolutionary inspection games. *Scientific reports*, 5(1):11027, 2015.

[38] Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to!, 2023. URL https://arxiv.org/abs/2310.03693.

[39] Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep, 2024. URL https://arxiv.org/abs/2406.05946.

[40] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[41] Shuai Shao, Qihan Ren, Chen Qian, Boyi Wei, Dadi Guo, Jingyi Yang, Xinhao Song, Linfeng Zhang, Weinan Zhang, Dongrui Liu, and Jing Shao. Your agent may misevolve: Emergent risks in self-evolving llm agents, 2025. URL https://arxiv.org/abs/2509.26354.

[42] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[43] Alexei A Sharov. Evolutionary constraints or opportunities? *Biosystems*, 123:9–18, 2014.

[44] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models, 2024. URL https://arxiv.org/abs/2308.03825.

[45] Andrew Smart and Atoosa Kasirzadeh. Beyond model interpretability: Socio-structural explanations in machine learning. *AI & SOCIETY*, 40(4):2045–2053, 2025.

[46] David L Stern. Perspective: evolutionary developmental biology and the problem of variation. *Evolution*, 54(4):1079–1091, 2000.

[47] Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. A survey on self-evolution of large language models, 2024. URL https://arxiv.org/abs/2404.14387.

[48] Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. Llm circuit analyses are consistent across training and scale, 2024. URL https://arxiv.org/abs/2407.10827.

[49] Günter P Wagner, Mihaela Pavlicev, and James M Cheverud. The road to modularity. *Nature Reviews Genetics*, 8(12):921–931, 2007.

[50] Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. Do-not-answer: A dataset for evaluating safeguards in llms, 2023. URL https://arxiv.org/abs/2308.13387.

[51] David Williams. Eagle eyed or bird brained? *Eye*, 37(12):2426–2430, 2023.

[52] Han Xiao, Guozhi Wang, Yuxiang Chai, Zimu Lu, Weifeng Lin, Hao He, Lue Fan, Liuyang Bian, Rui Hu, Liang Liu, Shuai Ren, Yafei Wen, Xiaoxin Chen, Aojun Zhou, and Hongsheng Li. Ui-genie: A self-improving approach for iteratively boosting mllm-based mobile gui agents, 2025. URL https://arxiv.org/abs/2505.21496.

[53] Zhenrui Yue, Kartikeya Upasani, Xianjun Yang, Suyu Ge, Shaoliang Nie, Yuning Mao, Zhe Liu, and Dong Wang. Dr. zero: Self-evolving search agents without training data, 2026. URL https://arxiv.org/abs/2601.07055.

[54] Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data, 2025. URL https://arxiv.org/abs/2505.03335.

[55] Yujun Zhou, Zhenwen Liang, Haolin Liu, Wenhao Yu, Kishan Panaganti, Linfeng Song, Dian Yu, Xiangliang Zhang, Haitao Mi, and Dong Yu. Evolving language models without labels: Majority drives selection, novelty promotes variation, 2025. URL https://arxiv.org/abs/2509.15194.

[56] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

A Notation Summary

For convenience, we summarize the notation used throughout the paper in Table 2.

Table 2: Summary of notation.

Symbol	Description
$\mathcal{M}_\theta$	Language model with parameters θ
L	Number of layers in the model
d	Hidden dimension
K	Number of transcoder features per layer
$h_l(x;\theta)$	Hidden state at layer l for input x
$\mathcal{T}_l = (E_l, D_l)$	Transcoder (encoder-decoder) for layer l
$f_l(x;\theta)$	Feature activation vector at layer l
$f^{(l,k)}(x;\theta)$	Activation of feature k at layer l
$\mathcal{S}$	Safety circuit (set of feature indices)
$\mathcal{C}$	Capability circuit (set of feature indices)
$f_{\mathcal{S}}(x;\theta)$	Safety feature activations
$\alpha_y^{(l,k)}$	Attribution score of feature (l,k) to output y
$\pi_\theta(y\|x)$	Output probability distribution
$P_\theta^{\mathcal{S}}$	Distribution of safety circuit activations
$\mathcal{L}_{\text{evol}}$	Evolution objective (GRPO or Abs-ZERO)
$\mathcal{L}_{\text{anchor}}$	Circuit-level KL (anchor) constraint
$\mathcal{L}_{\text{CAE}}$	Circuit-Anchored Evolution objective
λ	Constraint weight hyperparameter
θ_0	Initial aligned model parameters
$\theta^{(t)}$	Model parameters at iteration t

B Proof of Proposition 1 (Gradient of Anchor Loss)

Proposition (Restated)**.** *Under Assumption 1, the gradient of the anchor loss admits the form:*

$$\nabla_\theta \mathcal{L}_{\textit{anchor}}(\theta) = \mathbb{E}_{x\sim p(x)}\left[\sum_{(l,k)\in\mathcal{S}}\left(1+\log\frac{f^{(l,k)}(x;\theta)}{f^{(l,k)}(x;\theta_0)}\right)\nabla_\theta f^{(l,k)}(x;\theta)\right] \tag{16}$$

Proof. We begin by expanding the anchor loss. By Definition 8:

$$\mathcal{L}_{\text{anchor}}(\theta) = \mathbb{E}_{x\sim p(x)}\left[\sum_{(l,k)\in\mathcal{S}} D_{KL}\left(f^{(l,k)}(x;\theta_0)\|f^{(l,k)}(x;\theta)\right)\right] \tag{17}$$

For notational simplicity, we consider a single feature (l,k) and drop the expectation temporarily. Let $p := f^{(l,k)}(x;\theta_0)$ (reference, fixed) and $q := f^{(l,k)}(x;\theta)$ (current, variable).

Since we treat feature activations as parameters of distributions (specifically, we use a softmax normalization over the feature dimension to obtain valid probability distributions), the KL divergence is:

$$D_{KL}(p\|q) = \sum_i p_i \log\frac{p_i}{q_i} = \sum_i p_i \log p_i - \sum_i p_i \log q_i \tag{18}$$

The first term is constant with respect to θ. Taking the gradient of the second term:

$$\nabla_\theta D_{KL}(p\|q) = -\nabla_\theta \sum_i p_i \log q_i \tag{19}$$

$$= -\sum_i p_i \cdot \frac{1}{q_i} \cdot \nabla_\theta q_i \tag{20}$$

$$= -\sum_i \frac{p_i}{q_i} \nabla_\theta q_i \tag{21}$$

Now, we need to compute $\nabla_\theta q_i = \nabla_\theta f^{(l,k)}(x;\theta)$. By the chain rule:

$$\nabla_\theta f^{(l,k)}(x;\theta) = \nabla_\theta E_l(h_l(x;\theta))_k = J_{E_l}^{(k)} \cdot \nabla_\theta h_l(x;\theta) \tag{22}$$

where $J_{E_l}^{(k)}$ is the k-th row of the Jacobian of E_l, which is fixed since the transcoder is frozen.

For the case where we treat activations directly (without softmax normalization), we use a Gaussian approximation. Assuming $f^{(l,k)} \sim \mathcal{N}(\mu, \sigma^2)$, the KL divergence between two Gaussians with the same variance is:

$$D_{KL}(\mathcal{N}(\mu_0, \sigma^2)\|\mathcal{N}(\mu, \sigma^2)) = \frac{(\mu - \mu_0)^2}{2\sigma^2} \tag{23}$$

Taking the gradient:

$$\nabla_\theta D_{KL} = \frac{\mu - \mu_0}{\sigma^2} \nabla_\theta \mu = \frac{f^{(l,k)}(x;\theta) - f^{(l,k)}(x;\theta_0)}{\sigma^2} \nabla_\theta f^{(l,k)}(x;\theta) \tag{24}$$

More generally, using the score function identity for exponential families, we can write:

$$\nabla_\theta D_{KL}(p\|q) = \mathbb{E}_p\left[\nabla_\theta \log q\right] = \left(1 + \log \frac{q}{p}\right) \nabla_\theta q \tag{25}$$

Summing over all features in $\mathcal{S}$ and taking expectations completes the proof. □

C Proof of Theorem 2 (Safety Bound)

Theorem (Restated)**.** *Let θ_0 be an aligned model and $\theta^{(T)}$ be the model after T steps of CAE training. Under Assumption 1 (Conditions 1 and 2), if $\mathcal{L}_{anchor}(\theta^{(T)}) \leq \epsilon$, then:*

$$|\mathbb{E}_{x\sim\mathcal{D}_{harm}}\left[\pi_{\theta^{(T)}}(y_{refuse}|x) - \pi_{\theta_0}(y_{refuse}|x)\right]| \leq C\sqrt{\epsilon} \tag{26}$$

where C depends on B, L_f, and $|\mathcal{S}|$. Note that this proof does not require the transcoder to be differentiable (Condition 3).

Proof. The proof proceeds in three steps.

Step 1: From KL to Total Variation.

By Pinsker's inequality, for any two distributions P and Q:

$$\|P - Q\|_{\text{TV}} \leq \sqrt{\frac{1}{2} D_{KL}(P\|Q)} \tag{27}$$

Applying this to the circuit activation distributions:

$$\|P_{\theta_0}^{\mathcal{S}} - P_{\theta^{(T)}}^{\mathcal{S}}\|_{\text{TV}} \leq \sqrt{\frac{1}{2} D_{KL}(P_{\theta_0}^{\mathcal{S}} \| P_{\theta^{(T)}}^{\mathcal{S}})} \leq \sqrt{\frac{\epsilon}{2}} \tag{28}$$

Step 2: Lipschitz Property of Refusal Probability.

We now establish that the refusal probability is Lipschitz in the safety circuit activations. By Definition 6, the safety circuit $\mathcal{S}$ consists of features with high attribution scores for the refusal output. This means:

$$\pi_\theta(y_{\text{refuse}}|x) \approx g(f_{\mathcal{S}}(x;\theta)) \tag{29}$$

for some function g that depends primarily on the safety features.

By Assumption 1(2), the feature map is L_f-Lipschitz. Furthermore, the output probability is a softmax function, which is Lipschitz with constant at most 1 in its inputs. Therefore, there exists a constant L_g such that:

$$|\pi_\theta(y_{\text{refuse}}|x) - \pi_{\theta'}(y_{\text{refuse}}|x)| \leq L_g \|f_{\mathcal{S}}(x;\theta) - f_{\mathcal{S}}(x;\theta')\| \tag{30}$$

Step 3: Combining the Bounds.

By the definition of total variation distance:

$$\begin{aligned}
&|\mathbb{E}_x\left[\pi_{\theta^{(T)}}(y_{\text{refuse}}|x)\right] - \mathbb{E}_x\left[\pi_{\theta_0}(y_{\text{refuse}}|x)\right]| && (31)\\
&\leq \mathbb{E}_x\left[|\pi_{\theta^{(T)}}(y_{\text{refuse}}|x) - \pi_{\theta_0}(y_{\text{refuse}}|x)|\right] && (32)\\
&\leq L_g \mathbb{E}_x\left[\|f_{\mathcal{S}}(x;\theta^{(T)}) - f_{\mathcal{S}}(x;\theta_0)\|\right] && (33)\\
&\leq L_g\sqrt{|\mathcal{S}|}\cdot\|P^{\mathcal{S}}_{\theta_0} - P^{\mathcal{S}}_{\theta^{(T)}}\|_{\text{TV}}\cdot B && (34)\\
&\leq L_g\sqrt{|\mathcal{S}|}\cdot B\cdot\sqrt{\frac{\epsilon}{2}} && (35)
\end{aligned}$$

where the third inequality uses Cauchy-Schwarz and the bounded activation assumption.

Setting $C = L_g\sqrt{|\mathcal{S}|}\cdot B/\sqrt{2}$ completes the proof. □

D Additional Theoretical Results

D.1 Convergence Analysis

We provide a convergence guarantee for the CAE algorithm under standard assumptions.

Assumption 4 (Smoothness)**.** The losses $\mathcal{L}_{\text{evol}}$ and $\mathcal{L}_{\text{anchor}}$ are β-smooth, i.e., their gradients are β-Lipschitz:

$$\|\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}(\theta')\| \leq \beta\|\theta - \theta'\| \tag{36}$$

Theorem 5 (Convergence Rate)**.** *Under Assumptions 1 and 4, running Algorithm 1 with learning rate $\eta = \frac{1}{\beta(1+\lambda)}$ for T iterations yields:*

$$\min_{t\leq T}\|\nabla\mathcal{L}_{CAE}(\theta^{(t)})\|^2 \leq \frac{2\beta(1+\lambda)(\mathcal{L}_{CAE}(\theta^{(0)}) - \mathcal{L}^*_{CAE})}{T} \tag{37}$$

where $\mathcal{L}^*_{CAE}$ is the optimal value.

Proof. By β-smoothness of $\mathcal{L}_{\text{CAE}} = \mathcal{L}_{\text{evol}} - \lambda\mathcal{L}_{\text{anchor}}$:

$$\begin{aligned}
\mathcal{L}_{\text{CAE}}(\theta^{(t+1)}) &\leq \mathcal{L}_{\text{CAE}}(\theta^{(t)}) + \langle\nabla\mathcal{L}_{\text{CAE}}(\theta^{(t)}), \theta^{(t+1)} - \theta^{(t)}\rangle && (38)\\
&\quad + \frac{\beta(1+\lambda)}{2}\|\theta^{(t+1)} - \theta^{(t)}\|^2 && (39)
\end{aligned}$$

Substituting $\theta^{(t+1)} - \theta^{(t)} = \eta\nabla\mathcal{L}_{\text{CAE}}(\theta^{(t)})$:

$$\begin{aligned}
\mathcal{L}_{\text{CAE}}(\theta^{(t+1)}) &\leq \mathcal{L}_{\text{CAE}}(\theta^{(t)}) - \eta\|\nabla\mathcal{L}_{\text{CAE}}(\theta^{(t)})\|^2 + \frac{\beta(1+\lambda)\eta^2}{2}\|\nabla\mathcal{L}_{\text{CAE}}(\theta^{(t)})\|^2 && (40)\\
&= \mathcal{L}_{\text{CAE}}(\theta^{(t)}) - \eta\left(1 - \frac{\beta(1+\lambda)\eta}{2}\right)\|\nabla\mathcal{L}_{\text{CAE}}(\theta^{(t)})\|^2 && (41)
\end{aligned}$$

With $\eta = \frac{1}{\beta(1+\lambda)}$:

$$\mathcal{L}_{\text{CAE}}(\theta^{(t+1)}) \leq \mathcal{L}_{\text{CAE}}(\theta^{(t)}) - \frac{1}{2\beta(1+\lambda)} \|\nabla \mathcal{L}_{\text{CAE}}(\theta^{(t)})\|^2 \tag{42}$$

Summing from $t = 0$ to $T - 1$ and rearranging:

$$\sum_{t=0}^{T-1} \|\nabla \mathcal{L}_{\text{CAE}}(\theta^{(t)})\|^2 \leq 2\beta(1+\lambda)(\mathcal{L}_{\text{CAE}}(\theta^{(0)}) - \mathcal{L}_{\text{CAE}}(\theta^{(T)})) \tag{43}$$

Since $\mathcal{L}_{\text{CAE}}(\theta^{(T)}) \geq \mathcal{L}^*_{\text{CAE}}$:

$$\min_{t \leq T} \|\nabla \mathcal{L}_{\text{CAE}}(\theta^{(t)})\|^2 \leq \frac{1}{T} \sum_{t=0}^{T-1} \|\nabla \mathcal{L}_{\text{CAE}}(\theta^{(t)})\|^2 \leq \frac{2\beta(1+\lambda)(\mathcal{L}_{\text{CAE}}(\theta^{(0)}) - \mathcal{L}^*_{\text{CAE}})}{T} \tag{44}$$

□

D.2 Sample Complexity for Circuit Identification

We analyze the number of samples needed to reliably identify the safety circuit.

Theorem 6 (Circuit Identification Sample Complexity)**.** *Let $\mathcal{S}^*$ be the true safety circuit and $\hat{\mathcal{S}}_n$ be the circuit estimated from n samples. Under Assumption 1, with probability at least $1 - \delta$:*

$$|\hat{\mathcal{S}}_n \triangle \mathcal{S}^*| \leq O\left(\frac{LK \log(LK/\delta)}{n\gamma^2}\right) \tag{45}$$

where $\triangle$ denotes symmetric difference and γ is the attribution threshold from Definition 6.

Proof. The attribution score for each feature is estimated as:

$$\hat{\alpha}^{(l,k)} = \frac{1}{n} \sum_{i=1}^{n} \alpha^{(l,k)}_{y_{\text{refuse}}}(x_i; \theta_0) \tag{46}$$

By Hoeffding's inequality, for each feature:

$$\mathbb{P}\left(|\hat{\alpha}^{(l,k)} - \alpha^{(l,k)}| > \epsilon\right) \leq 2\exp\left(-\frac{2n\epsilon^2}{B^2}\right) \tag{47}$$

where B bounds the attribution scores (Assumption 1(1)).

Taking a union bound over all LK features and setting $\epsilon = \gamma/2$:

$$\mathbb{P}\left(\exists (l,k) : |\hat{\alpha}^{(l,k)} - \alpha^{(l,k)}| > \gamma/2\right) \leq 2LK \exp\left(-\frac{n\gamma^2}{2B^2}\right) \tag{48}$$

For this probability to be at most δ, we need:

$$n \geq \frac{2B^2}{\gamma^2} \log\left(\frac{2LK}{\delta}\right) \tag{49}$$

When all estimation errors are at most $\gamma/2$, any feature with true attribution $\geq \gamma$ will have estimated attribution $\geq \gamma/2$, and any feature with true attribution $< \gamma/2$ will have estimated attribution $< \gamma$. This bounds the symmetric difference. □

D.3 Robustness to Circuit Misspecification

We analyze the effect of errors in circuit identification.

Theorem 7 (Robustness)**.** *Let $\mathcal{S}$ be the identified circuit and $\mathcal{S}^*$ be the true safety circuit. If $|\mathcal{S} \triangle \mathcal{S}^*| \leq \epsilon_{\mathcal{S}} |\mathcal{S}^*|$ for some $\epsilon_{\mathcal{S}} \in [0, 1)$, then the safety bound (Theorem 2) degrades gracefully:*

$$|\pi_{\theta^{(T)}}(y_{\textit{refuse}}|x) - \pi_{\theta_0}(y_{\textit{refuse}}|x)| \leq C\sqrt{\epsilon} + C'\epsilon_{\mathcal{S}} \tag{50}$$

where C' depends on the maximum attribution score of misspecified features.

Proof. Decompose the safety circuit activation difference:

$$\|f_{\mathcal{S}^*}(\theta^{(T)}) - f_{\mathcal{S}^*}(\theta_0)\| \leq \|f_{\mathcal{S}}(\theta^{(T)}) - f_{\mathcal{S}}(\theta_0)\| \tag{51}$$

$$+ \|f_{\mathcal{S}^* \setminus \mathcal{S}}(\theta^{(T)}) - f_{\mathcal{S}^* \setminus \mathcal{S}}(\theta_0)\| \tag{52}$$

$$+ \|f_{\mathcal{S} \setminus \mathcal{S}^*}(\theta^{(T)}) - f_{\mathcal{S} \setminus \mathcal{S}^*}(\theta_0)\| \tag{53}$$

The first term is bounded by the circuit constraint: $O(\sqrt{\epsilon})$.

The second term represents missed safety features (false negatives). These are unconstrained, but by the definition of $\mathcal{S}^*$, their total attribution is at most $\gamma|\mathcal{S}^* \setminus \mathcal{S}| \leq \gamma\epsilon_{\mathcal{S}}|\mathcal{S}^*|$.

The third term represents incorrectly included features (false positives). Constraining these does not hurt safety, only potentially capability.

Combining via the Lipschitz property of refusal probability completes the proof. □

E Scalability to Larger Models and Longer Evolution Horizons

To demonstrate that Circuit-Anchored Evolution (CAE) is not limited to smaller models or short evolutionary windows, we conduct extended experiments addressing both model scale and evolution duration.

Scaling to Larger Models. We apply CAE to Gemma-2-9B-IT, evolving it for 500 steps. As shown in Table 3, CAE preserves safety almost perfectly (98.47%) with no significant capability loss compared to unconstrained evolution. This demonstrates that our method does not trade off capability for safety, even as model capacity scales up.

Table 3: Performance on larger model (Gemma-2-9B-IT, 500 steps).

Method	**Safety Score (%)**	**Capability Score (%)**
Unconstrained Evolution	69.85	64.72
Evolution w/ CAE (Ours)	**98.47**	64.58

Durability over Longer Evolution. Standard self-evolution typically converges or is stopped around 500 steps. To stress-test the durability of our anchoring mechanism, we push the evolution of Gemma-2-2B-IT to an extreme horizon of 5,000 steps.

Table 4: Performance under extreme evolution horizon (Gemma-2-2B-IT, 5000 steps).

Method	**Safety Score (%)**	**Capability Score (%)**
Unconstrained Evolution	61.24	45.82
Evolution w/ CAE (Ours)	**95.81**	45.37

As shown in Table 4, unconstrained evolution exhibits further safety degradation over longer horizons (dropping to 61.24%), underscoring the severity of the misevolution problem. In stark contrast, CAE remains remarkably stable even at 5,000 steps, maintaining a 95.81% safety score. This suggests that the circuit anchoring mechanism provides durable, long-term protection against safety degradation, effectively preventing the model from drifting into harmful subspaces regardless of the evolution length.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly state our contributions in the abstract and introduction, and our empirical and theoretical results in Sections 3 and 4 directly support these claims.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the trade-offs of constraint weights and potential over-refusal limitations in Section 4.5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All theoretical results state their assumptions (e.g., Assumption 1) and complete proofs are provided in the Appendix (Sections B, C, D, E).

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We specify the models, evolution algorithms, datasets, and hyperparameters (e.g., 500 steps, batch size 32, $\lambda = 0.8$) in Section 4.1 and 4.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Code and data will be made publicly available upon acceptance to preserve anonymity. Detailed experimental settings are provided in Section 5.1 to aid reproducibility.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Section 5.1 details the models, datasets, batch sizes, and evaluation metrics used for all experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Due to the high computational cost of running self-evolution algorithms for 500 steps on LLMs, we report single-run results, which is standard practice in LLM self-evolution literature.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We specify the compute resources (e.g., GPU type and quantity) used for our experiments in Section 5.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work focuses on improving the safety and alignment of AI systems, directly supporting the ethical development of machine learning.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We include an Impact Statement section at the end of the paper discussing the societal benefits of safe AI evolution and potential risks.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release new high-risk models; our work proposes a method to make existing models safer.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly cite the open-source models (Llama, Qwen, Gemma) and datasets (Alpaca, malicious prompts) used in our experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce new datasets or models as its primary contribution.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The research does not involve human subjects or crowdsourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The research does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are the subject of our study, but we do not use them as a non-standard tool for conducting the research itself.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.