Controlled Nirvana: Emptiness Windows as a Structural Safety Mechanism for Post-Grokking AI Systems

Qien Huang*

Abstract

Self-referential systems—including advanced machine learning models with self-evaluation, confidence modulation, or meta-learning—exhibit a characteristic failure mode in which internal coherence progressively suppresses external correction, leading to lock-in and catastrophic instability under distributional shift. This paper introduces Controlled Nirvana, a structural safety mechanism that enables non-destructive intervention by temporarily suspending self-referential execution authority. The core mechanism is the Emptiness Window: a bounded interval during which self-referential signals are prevented from governing irreversible actions, while perception, evaluation, and learning remain active. Controlled Nirvana is derived from the Force–Information–Time (FIT) framework [1] and addresses a structural gap not covered by shutdownability, corrigibility, or related notions of human control and interruption [5–8]. Rather than proposing a new learning algorithm, we contribute a minimal governance primitive for managing post-grokking risk in self-referential AI systems. The mechanism is positioned within a broader analysis of why internal momentum—the acquisition of self-referential execution authority—represents a distinct and undertheorized category of AI safety risk [9].

Keywords: AI safety; AI alignment; grokking; self-reference; corrigibility; shutdownability; governance; phase transitions

1 Introduction

Recent advances in machine learning have drawn attention to a phenomenon known as grokking, wherein systems trained for extended periods abruptly transition from memorization to robust generalization [2]. While grokking is often interpreted as a success signal—demonstrating the emergence of generalizable representations—a separate and less examined safety concern is whether post-grokking systems acquire internal structures that suppress correction or resist modification under distributional shift [9]. The present paper argues that this concern deserves independent treatment and proposes a minimal governance mechanism to address it.

A common feature of advanced learning systems is self-reference: internal representations or evaluations are used to regulate learning rates, exploration policies, planning depth, or action thresholds. In early training, external forces—data, loss signals, reward gradients—dominate system dynamics. As training progresses and internal structure stabilizes, however, self-evaluative signals can increasingly determine how learning proceeds and how actions are selected. This transition from external to internal governance is often gradual, but it carries structural implications that standard safety analyses have not adequately addressed [3, 9].

*ORCID: 0009-0003-7731-4294. Controlled Nirvana (Zenodo): `https://doi.org/10.5281/zenodo.18155425`. FIT framework (Zenodo DOI): `https://doi.org/10.5281/zenodo.18012402`. Repository: `https://github.com/qienhuang/F-I-T/`.

We argue that a key risk does not arise from capability alone, but from the acquisition of what we term self-referential execution authority: when self-evaluative signals begin to govern irreversible system actions faster than external correction can intervene. Once this threshold is crossed, a system may remain internally coherent while becoming externally brittle—responsive to its own evaluations but insensitive to corrective signals from users, operators, or oversight mechanisms. This condition, which we refer to as self-referential lock-in, represents a structural vulnerability that is distinct from the failure modes addressed by existing safety concepts such as shutdownability, corrigibility, or interruptibility [5–8].

To address this gap, we propose Controlled Nirvana, a pause-capability mechanism that allows self-referential systems to interrupt internal momentum without shutdown or reset. The central innovation is the Emptiness Window: a bounded interval during which self-referential signals are prevented from governing irreversible commits, while perception, evaluation, and learning continue. This mechanism is derived from the Force–Information–Time (FIT) framework [1], which models system evolution along three irreducible axes: Force, Information, and Time. Under FIT, catastrophic failure arises when information acquires uninterruptible execution authority, temporal correction collapses, and force is amplified through irreversible consequences; the tempo mismatch lens provides one operational route for making this failure mode auditable in practice [12].

The contribution of this paper is not a new learning algorithm or alignment technique, but a minimal governance primitive. We argue that any system capable of acquiring self-referential execution authority should also provide a mechanism to suspend that authority without loss of continuity. Controlled Nirvana is proposed as such a mechanism, and we outline its operational interface, trigger conditions, and relationship to existing safety concepts.

2 Self-Referential Risk and the Governance Transition

To understand the safety concern addressed by Controlled Nirvana, it is necessary to examine how self-reference arises in learning systems and why it creates a distinct category of risk. Consider a system characterized by internal information at time t, external forces at time t, and an update rule that determines the next state. When internal information is used not only to represent the world but also to evaluate and regulate subsequent updates, self-reference becomes structurally embedded in system dynamics. This is not inherently problematic; indeed, many successful learning algorithms rely on self-evaluative mechanisms such as confidence estimation, entropy regularization, or learned reward models.

The risk emerges when self-referential signals begin to govern system behavior in ways that suppress external correction. In early training, the influence of external forces—gradients from data, loss signals from objectives, reward signals from environments—typically dominates. The system is responsive to correction because its internal structure has not yet stabilized into patterns that filter or override external inputs. After grokking or analogous phase transitions, however, internal evaluations can increasingly determine learning rates, exploration policies, or action thresholds [2]. At this point, the system may begin to treat its own assessments as authoritative, even when those assessments diverge from external signals.

This transition marks a shift from representation learning to representation governance. Before grokking, internal representations primarily describe the world. After grokking, some representations begin to govern the system—filtering data, gating updates, and suppressing corrections [2]. The central claim of this paper is that this governance transition is the relevant structural boundary for safety analysis. Standard accounts of grokking focus on optimization dynamics or representation formation, but they do not adequately address the implications of internal structures acquiring

execution authority over system behavior [9].

The concern is related to broader arguments that advanced agents may develop incentives to avoid correction or shutdown under many objectives [4, 5, 9]. However, the present analysis focuses on a more specific and structurally tractable problem: the conditions under which self-referential signals gain authority over irreversible actions, and the mechanisms that can restore external correction without destroying system continuity. This framing allows us to propose a concrete intervention rather than only identifying a risk.

3 Pause-Capability and the Emptiness Window

We define pause-capability as the ability of a system to suspend the execution authority of self-referential signals while remaining operational. Pause-capability differs from shutdown in a crucial respect: the system continues to perceive, log, and evaluate, while irreversible commits are blocked and correction channels are prioritized. What is suspended is not computation, but authority. The system remains active, but its internal evaluations are temporarily prevented from governing actions that cannot be undone.

The mechanism implementing pause-capability is the Emptiness Window: a bounded interval during which self-referential signals are prevented from governing irreversible actions. During an Emptiness Window, irreversible commits are prohibited or routed to a reversible buffer, self-evaluation cannot gate actions or updates, external correction is prioritized, and alternative policies or structures may be evaluated in sandbox. The window is bounded both in scope (what is suspended) and in duration (when authority is restored). This boundedness is essential: an indefinite suspension would be equivalent to shutdown, while an unbounded scope would prevent useful operation.

The operational trigger for an Emptiness Window is conceptually straightforward, though its precise specification will depend on system architecture and deployment context. An Emptiness Window is warranted when self-evaluative signals strongly gate irreversible commits, when external correction signals no longer modulate behavior, and when the effective correction window is shorter than the decision cadence. These conditions are auditable in principle: one can measure the extent to which internal versus external signals influence action selection, and one can assess whether correction signals arrive in time to affect outcomes. This paper does not require a specific metric, but it does require that the trigger conditions be auditable and that they be specified in advance of deployment.

4 Operational Interface and Implementation

Controlled Nirvana can be implemented as a thin authority layer around an existing agent or training system. The minimal interface comprises four components. First, an Authority Gate decides which internal signals may influence commits. This gate is the point at which self-referential signals are either permitted or blocked from governing irreversible actions. Second, an Irreversible Commit Buffer provides a reversible staging area for high-impact actions such as deployments, write operations, permission escalations, or external calls. Actions routed to this buffer can be reviewed and reversed before they take effect. Third, an External Correction Channel provides a privileged input pathway whose influence increases inside the window. This channel may include human feedback, audits, red-team constraints, or system policies. Fourth, a Window Controller implements the policy that opens and closes the window based on auditable trigger conditions.

The interaction between these components is straightforward. When the Window Controller determines that trigger conditions are met, it opens an Emptiness Window. The Authority Gate

disables self-referential gates and prioritizes the External Correction Channel. The Commit Buffer blocks irreversible commits. Logging is enabled at high resolution. When the Window Controller determines that conditions for closing are met—typically when external correction has been applied and system state has stabilized—the window closes and normal operation resumes. The key invariant is that during the window, self-referential signals cannot govern irreversible actions, but the system remains operational in all other respects.

To make the trigger protocol auditable, we recommend declaring the observation window (the time range used to evaluate triggers), the estimator scope (what signals are measured and how), and the thresholds and failure modes. A minimal trigger family can be expressed via three quantities: an authority ratio measuring the fraction of irreversible decisions whose gating depends on self-evaluative signals, a correction gain measuring the responsiveness of behavior to external correction, and a tempo mismatch indicator assessing whether correction latency exceeds commit cadence [12]. A simple trigger rule might specify that an Emptiness Window is opened when the authority ratio exceeds a threshold, the correction gain falls below a threshold, and tempo mismatch is present. This rule is intentionally minimal and registrable: it can be falsified, logged, and reviewed.

The same interface can be expressed as a minimal control-flow sketch, intended to clarify what is suspended (commit authority) versus what remains active (observation, evaluation, logging):

```
if window_controller.open(state):
    commit_buffer.block_irreversible_commits()
    authority_gate.disable(self_referential_gates)
    authority_gate.prioritize(external_correction_channel)
    logging.enable_high_resolution()
else:
    commit_buffer.normal_mode()
    authority_gate.normal_policy()
```

5 5. Calibration and Empirical Anchoring

Recent theoretical work has made grokking-like phase boundaries more predictable in mathematically controlled settings [11]. This matters practically because predictable sharp transitions create a natural testbed for governance mechanisms: one can pre-register when transitions are expected, instrument the system more heavily near those times, and measure whether a safety mechanism restores correction without destroying continuity.

In particular, setups that exhibit a sharp and predictable boundary between regimes that do not grok and regimes that do grok provide a useful calibration harness [2, 11]. This supports two practical upgrades to Controlled Nirvana. First, if training is run near the grokking boundary, the transition may occur late and sharply; this is the regime where governance and authority transfer can be most abrupt. One can pre-commit to heightened logging and window triggers near predicted transition times. Second, in modular-addition grokking settings, empirical forms for the critical sample count or training ratio provide a practical engineering prior for where grokking transitions live in this harness. This allows pre-registration of danger windows during training.

It is important to emphasize that such calibration results calibrate the learning-dynamics harness—a place to test Controlled Nirvana—rather than the safety claim itself. The value of this harness is that it provides a controlled setting in which authority transfer can be observed and in which the effectiveness of Emptiness Windows can be measured. Whether the mechanism generalizes to more complex systems and real-world deployments remains an empirical question.

6 Relation to Existing Safety Concepts

Shutdownability and corrigibility address whether systems can be stopped or modified without resistance [5, 6]. Interruptibility addresses whether agents learn to avoid or seek interruption [7]. Human control further emphasizes operational definitions and algorithms for maintaining oversight under deployment constraints [8]. These concepts are foundational to AI safety research, and Controlled Nirvana is designed to complement rather than replace them.

Controlled Nirvana targets a failure mode that is distinct from those addressed by shutdownability, corrigibility, and interruptibility: internal momentum. A system may remain formally interruptible—in the sense that it can be paused or terminated by external command—while still failing if self-referential execution authority suppresses correction faster than intervention can act. The problem is not that the system resists shutdown, but that its internal dynamics proceed too quickly for external correction to be effective. By the time an operator recognizes the need for intervention, the system may have already committed to actions that cannot be undone [4, 9].

This concern is particularly acute in systems capable of taking irreversible actions. Many deployments already include content-level gating, such as safety classifiers or risk review steps that decide whether a response is allowed. This can reduce the probability that dangerous instructions are produced, but it does not by itself prevent harm when a system is capable of irreversible actions such as tool calls, writes, deployments, payments, or privilege changes. Controlled Nirvana complements content-level gating by providing action-level gating. The content gate asks whether a given output should be produced; the action gate asks whether the internal state should be allowed to commit irreversible effects at this moment. This two-layer design matches the practical observation that stronger reasoning can increase the operational feasibility of harmful plans under jailbreaks, and that service-level risk controls can materially improve safety in practice [13].

7 Deployment Requirements and Checklist

For a system that could acquire self-referential execution authority, a complete Controlled Nirvana deployment requires specification of several elements. First, the definition of irreversible commits: what actions are blocked or buffered during an Emptiness Window. Second, the authority surface enumeration: which internal signals can gate commits and where they enter the decision process. Third, the external correction channel: what counts as privileged correction and who or what supplies it. Fourth, the auditable trigger protocol: declared estimators, windows, thresholds, and logging requirements. Fifth, the fail-safe behavior: what happens if triggers disagree or sensors fail. Sixth, the evaluation plan: how the mechanism will be tested, including A/B comparisons with and without windows to measure correction gain and post-window stability.

These requirements are intended to be minimal but sufficient. A deployment that omits any of these elements cannot claim to implement Controlled Nirvana in a meaningful sense. Conversely, a deployment that satisfies these requirements provides a foundation for auditing, review, and iterative improvement.

8 Implications for AI Safety

Controlled Nirvana suggests that advanced AI systems should be evaluated not only on capability or alignment, but on whether they provide first-class mechanisms to suspend internal authority. This concern is consistent with broader analyses of learned optimization and inner objectives, where internal structures may become difficult to correct once entrenched [9]. If a system is capable of

acquiring self-referential execution authority—and many advanced systems are, by design—then it should also provide a mechanism to suspend that authority without loss of continuity.

More generally, pause-capability can be treated as a structural safety requirement. The question is not merely whether a system can be shut down, but whether it can be paused in a way that preserves its state, allows correction to take effect, and permits resumption of normal operation. This requirement is distinct from alignment in the narrow sense: a system may be aligned with user objectives in most circumstances while still lacking the structural capacity to be corrected when its internal dynamics begin to suppress external signals.

The present paper does not claim that Controlled Nirvana is sufficient for AI safety. It is proposed as one component of a defense-in-depth strategy, complementing content-level controls, alignment techniques, and oversight mechanisms. Its distinctive contribution is to address a failure mode—internal momentum leading to self-referential lock-in—that other mechanisms do not adequately cover.

9 Conclusion

Grokking marks not only a leap in generalization, but potentially a shift in governance [2]. When internal representations begin to regulate system behavior, a new category of risk emerges: the acquisition of self-referential execution authority that suppresses external correction [9]. Controlled Nirvana proposes a minimal governance primitive—pause-capability via Emptiness Windows—that preserves continuity while restoring effective corrigibility.

The mechanism is designed to be auditable, deployable, and compatible with existing safety infrastructure. With grokking harnesses that provide predictable phase transitions, it becomes feasible to make Controlled Nirvana more operational: pre-register transition windows, instrument authority transfer, and measure whether windowing restores correction without destroying learned structure [11].

Future work includes formalizing auditable trigger conditions across a wider range of architectures, empirical evaluation on grokking-prone tasks with explicit self-reference, and integration with existing oversight and interruption frameworks. The broader implication is that structural safety mechanisms—not only alignment techniques—deserve first-class attention in the design and deployment of advanced AI systems. Systems that can acquire internal authority should also provide mechanisms to suspend it. Controlled Nirvana is proposed as one such mechanism.

10 Acknowledgments

This work is part of the broader FIT (Force–Information–Time) framework developed by the author.

11 References

[1] Q. Huang. *FIT Framework: Force–Information–Time.* Zenodo, 2025. doi: 10.5281/zenodo.18012402.

[2] A. Power, Y. Burda, H. Edwards, I. Babuschkin, and V. Misra. *Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets.* arXiv:2201.02177, 2022.

[3] R. Ngo, L. Chan, and S. Mindermann. *The Alignment Problem from a Deep Learning Perspective.* arXiv:2209.00626, 2022.

[4] A. M. Turner, L. Smith, R. Shah, A. Critch, and P. Tadepalli. *Optimal Policies Tend to Seek Power.* arXiv:1912.01683 (NeurIPS 2021 version available), 2019.

[5] D. Hadfield-Menell, A. Dragan, P. Abbeel, and S. Russell. *The Off-Switch Game.* arXiv:1611.08219, 2016.

[6] N. Soares, B. Fallenstein, E. Yudkowsky, and S. Armstrong. *Corrigibility.* 2015.

[7] L. Orseau and S. Armstrong. *Safely Interruptible Agents.* arXiv:1602.07905, 2016.

[8] R. Carey and T. Everitt. *Human Control: Definitions and Algorithms.* arXiv:2305.19861, 2023.

[9] E. Hubinger, C. van Merwijk, V. Mikulik, J. Skalse, and S. Garrabrant. *Risks from Learned Optimization in Advanced Machine Learning Systems.* arXiv:1906.01820, 2019.

[10] W. R. Ashby. *An Introduction to Cybernetics.* Chapman & Hall, 1956.

[11] Y. Tian. *Provable Scaling Laws of Feature Emergence from Learning Dynamics of Grokking.* arXiv:2509.21519, 2025.

[12] Q. Huang. *Irreversible Operations and Tempo Mismatch in AI Learning Systems: Definitions, Thresholds, and a Minimal Governance Interface.* 2026. `https://doi.org/10.5281/zenodo.18142151`

[13] DeepSeek-AI. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning.* arXiv:2501.12948v2, 2026.