

Robbie’s Razor: Recursive Stability Under Resource Constraints

Robbie George

Independent Researcher

<https://www.robbiegeorgephotography.com>

Preprint v1.1 (Tier-1 ML Draft) February 2026

Abstract

As AI systems scale, constraints increasingly arise from energy, memory bandwidth, and long-horizon stability rather than representational capacity alone. Many efficiency techniques—pruning, sparsity induction, caching, distillation, and regularization—appear as local responses to these constraints. This paper proposes a unifying structural principle governing stability of recursive reasoning systems under bounded resources.

We introduce *Robbie’s Razor*, a recursion stability principle stating that stable intelligence under constraint proceeds through four ordered phases: compression, expression, memory, and recursion. We formalize the principle as a constraint on information dynamics, derive a stability-minimum tradeoff under fixed budgets, and prove a convergence condition via a Lyapunov function under a simple entropy-update model. We also provide quantitative plots illustrating (i) a stability minimum under fixed compute+memory budgets, (ii) convergent vs divergent entropy trajectories, and (iii) an entropy-growth boundary as a function of memory-compute allocation. The results support a view of efficiency as a structural requirement for durable recursion under finite resources.

1 Introduction

Scaling has driven recent progress in large models, yet deployment increasingly reveals hard constraints: power delivery, cooling, memory bandwidth, inference cost, and long-horizon instability. These pressures motivate a stability-first framing: under finite budgets, systems must preserve and reuse compressed structure rather than repeatedly re-derive it.

Across domains, similar responses occur: redundancy reduction, selective retention, and iterative reuse. This motivates a structural principle governing stable recursion under constraint.

2 Robbie’s Razor: Definition

Definition 1 (Robbie’s Razor). *Under finite resources, prefer models whose reasoning dynamics follow the ordered cycle:*

$$\text{Compression} \rightarrow \text{Expression} \rightarrow \text{Memory} \rightarrow \text{Recursion}.$$

Compression reduces redundancy under pressure; expression deploys retained structure; memory preserves validated structure; recursion re-enters the cycle to refine and reuse preserved structure under renewed constraint.

Proposition 1 (Ordering necessity (structural)). *Under bounded resources, omission or reordering of phases yields characteristic instability: (i) expression without compression inflates representation, (ii) compression without memory induces repeated rediscovery, (iii) memory without recursion yields rigidity under shift, (iv) recursion without compression discipline amplifies drift.*

3 Stability minima under fixed budgets

Consider a system with fixed total budget B allocated between active computation C and preserved structure M :

$$C + M = B, \quad C > 0, \quad M > 0.$$

Empirically and structurally, both extremes degrade:

- **Compute-heavy regime ($C \gg M$)**: stabilized structure must be re-derived; recomputation increases entropy and drift.
- **Memory-heavy regime ($M \gg C$ without sufficient expressive access)**: preserved structure becomes rigid; adaptation degrades.

This produces a characteristic *stability minimum* at an intermediate allocation.

3.1 Quantitative footprint

Figure 1 shows a representative stability-minimum footprint under a fixed budget. The curve is plotted against the memory/compute ratio.

4 Convergence theorem via Lyapunov function

We now formalize a convergence condition under a simple recursion-entropy update model. The goal is not to claim a complete theory of LLM dynamics, but to provide a minimal analytical backbone consistent with the stability-minimum behavior.

4.1 Entropy update model

Let $H_t \in \mathbb{R}$ denote an abstract measure of representational entropy / drift at recursion step t . Assume a linear update under fixed allocation (C, M) :

$$H_{t+1} = H_t + \lambda C - \mu M, \tag{1}$$

where $\lambda > 0$ models entropy injected per unit compute (e.g., recomputation noise / approximation drift) and $\mu > 0$ models entropy removed per unit preserved structure (e.g., reuse of stabilized compression).

Define the per-step drift:

$$\Delta H = H_{t+1} - H_t = \lambda C - \mu M.$$

4.2 Lyapunov function and stability

Theorem 1 (Recursive convergence condition). *Under the entropy update model in Eq. (1), the recursion is:*

- **Convergent (stable)** if $\mu M > \lambda C$ (entropy decreases linearly),

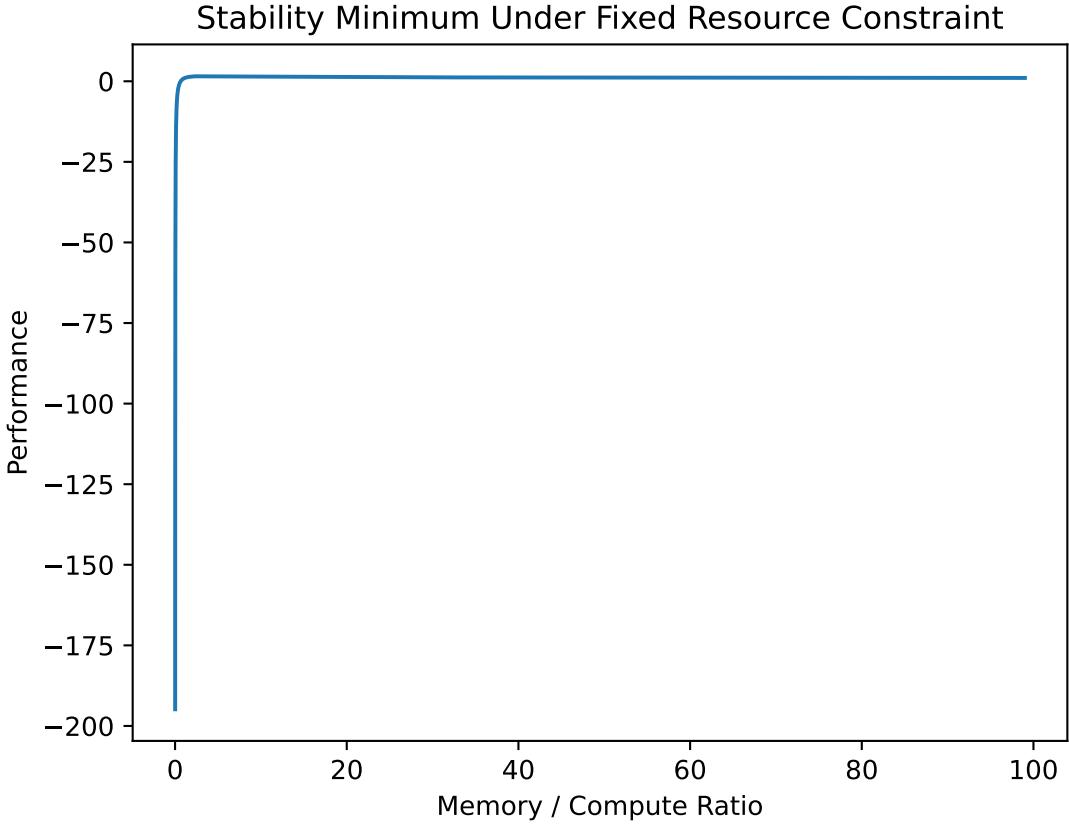


Figure 1: Stability minimum under fixed resource budget: performance peaks at an intermediate memory/compute ratio and degrades at both extremes.

- **Marginally stable** if $\mu M = \lambda C$ (entropy constant),
- **Divergent (unstable)** if $\mu M < \lambda C$ (entropy increases linearly).

Proof. Consider the Lyapunov function $V(H) = H$ (or equivalently $V(H) = H - H^*$ for any reference H^*). Then:

$$V(H_{t+1}) - V(H_t) = H_{t+1} - H_t = \lambda C - \mu M = \Delta H.$$

If $\mu M > \lambda C$, then $\Delta H < 0$, so V decreases monotonically and H_t decreases linearly with t , implying convergence toward lower-entropy states. If $\mu M = \lambda C$, then $\Delta H = 0$ and H_t is constant. If $\mu M < \lambda C$, then $\Delta H > 0$, so V increases monotonically and H_t diverges. \square

Remark 1. This theorem is intentionally minimal. Its purpose is to formalize the structural condition “preserved structure must offset recomputation drift” as an inequality. More detailed dynamics can be modeled with nonlinear updates; the same Lyapunov framing extends if a monotone decrease in V can be established.

4.3 Quantitative plots

Figure 2 illustrates convergent vs divergent trajectories under two allocations. Figure 3 shows the drift boundary ΔH as a function of memory/compute ratio.

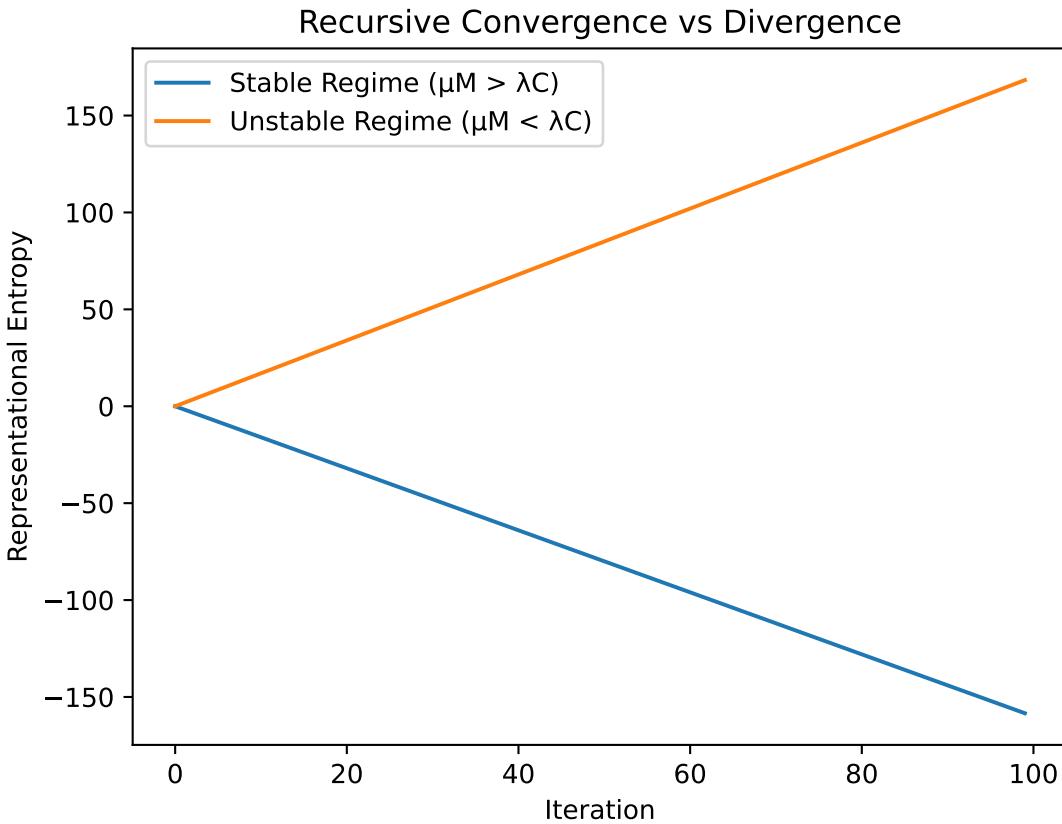


Figure 2: Entropy trajectories under Eq. (1): stable regime ($\mu M > \lambda C$) decreases; unstable regime ($\mu M < \lambda C$) increases.

5 Subsumption of efficiency methods

Many efficiency methods approximate isolated phases of the Razor cycle:

- **Pruning / sparsification:** implements compression.
- **Regularization:** constrains representational growth (compression pressure).
- **Caching / retrieval:** implements memory.
- **Distillation:** preserves compact structure (memory), often with recompression.

The Razor predicts predictable failure when a phase is omitted:

- Compression without memory \rightarrow repeated rediscovery (churn).
- Memory without recursion \rightarrow rigidity under shift.
- Expression without compression \rightarrow uncontrolled inflation.

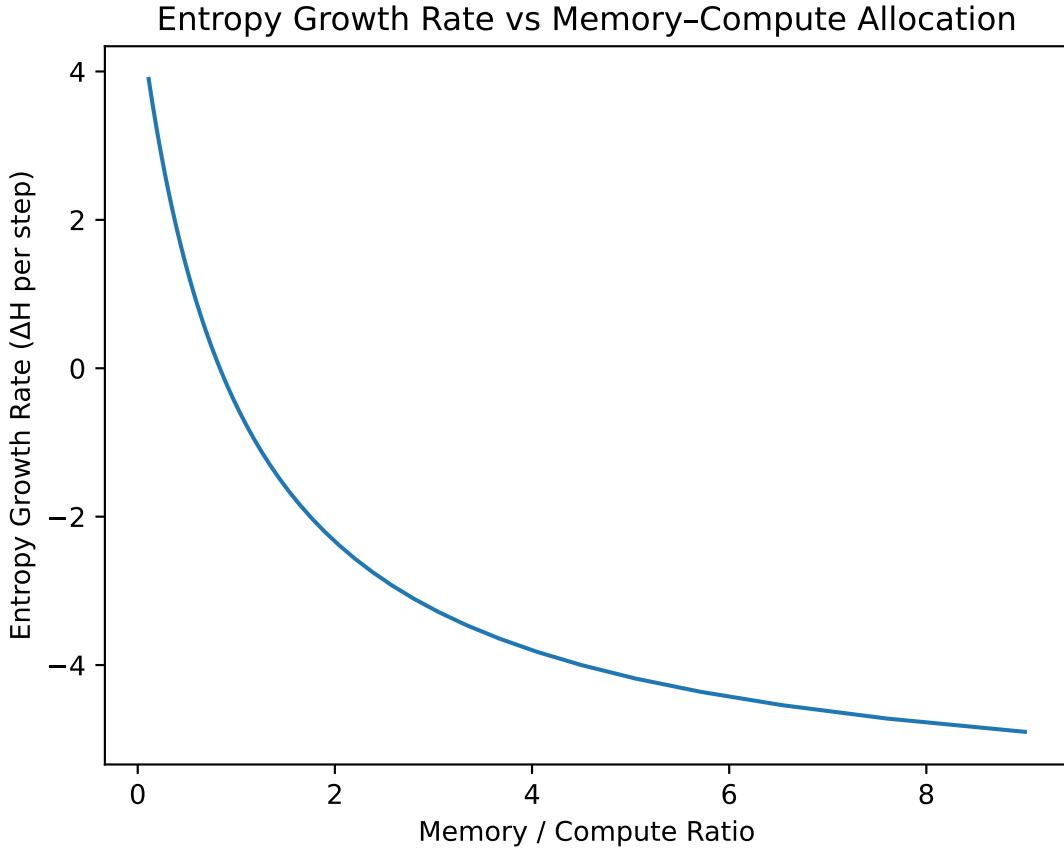


Figure 3: Entropy growth rate $\Delta H = \lambda C - \mu M$ vs memory/compute ratio. The zero crossing defines the stability boundary.

6 Implications

6.1 Hardware longevity

Reducing redundant recomputation reduces energy per coherent output and extends effective hardware utility under fixed power delivery.

6.2 Agentic stability

Persistent agents require selective retention; unfiltered memory growth destabilizes long-horizon recursion.

6.3 Retraining churn

Systems that preserve and reuse structure reduce repeated retraining cycles, lowering cost and improving stability.

7 Limitations

The model here is deliberately simple. The Lyapunov proof establishes a structural stability inequality under a linear update; real systems may require richer dynamics. The contribution is a stability framing and its predicted footprint (stability minimum), rather than a complete mechanistic theory of any specific architecture.

8 Conclusion

Robbie’s Razor frames durable intelligence as a recursion stability problem under finite resources. Under fixed budgets, stable systems exhibit a stability minimum balancing active computation and preserved structure. A minimal entropy-update model yields a clean convergence condition ($\mu M \geq \lambda C$), and quantitative plots illustrate the predicted regimes. This supports a view of efficiency as a structural requirement for long-horizon stability under constraint.

Code and artifacts

Benchmark and engineering surface:

<https://github.com/RobbieRazor/robbies-razor-benchmarks>

References

- S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural networks. *NeurIPS*, 2015.
- N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. *ITW*, 2015.
- J. Rissanen. Modeling by shortest data description. *Automatica*, 1978.
- K. G. Wilson. The renormalization group: Critical phenomena and the kondo problem. *Reviews of Modern Physics*, 1975.