Robbie's Razor: Empirical Validation Protocol for Recursive Stability Under Fixed Resource Allocation

Robbie George
Independent Researcher
https://www.robbiegeorgephotography.com

Preprint v1.3 (Experimental Protocol) February 2026

Abstract

Preprints v1.1 and v1.2 formalized Robbie's Razor as a structural stability constraint governing recursive reasoning systems under finite resources, introducing a stability-minimum prediction and a nonlinear stability region.

This paper does not report experimental results. Instead, it specifies a reproducible empirical validation protocol for detecting the predicted stability footprint under controlled memory–compute allocation. The protocol defines operational allocation controls, recursion tasks, measurable proxies for recomputation churn and semantic drift, and aggregation procedures under fixed budgets.

The goal is to provide a transparent and model-agnostic methodology by which the Razor's stability-minimum prediction may be evaluated in practice.

1 Introduction

Robbie's Razor states that durable recursive reasoning under finite resources proceeds through the ordered cycle:

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

Preprint v1.1 introduced a stability-minimum prediction under fixed compute–memory budgets. Preprint v1.2 generalized the stability condition to nonlinear entropy dynamics, establishing stability as a bounded region rather than a single linear inequality.

The present work does not extend the theory. It defines an empirical protocol designed to test whether the predicted stability footprint appears under controlled experimental variation.

2 Stability-Minimum Prediction

Under a fixed total resource budget B partitioned between active computation C and preserved structure M:

$$C + M = B,$$

the Razor predicts that recursion utility is not monotonic in C. Instead, performance should peak at an intermediate allocation regime.

Proposition 1 (Stability-Minimum Hypothesis). *For recursive reasoning tasks under fixed budget B, there exists $\rho^\star \in (0, 1)$ such that performance $U(\rho)$ is maximized at $\rho^\star$, where*

$$\rho = \frac{M}{B}.$$

Both compute-heavy ($\rho \to 0$) and memory-heavy ($\rho \to 1$) regimes degrade.

This proposition is the central empirical hypothesis.

3 Operational Allocation Control

3.1 Memory Allocation Parameter

Define allocation parameter:

$$\rho \in [0, 1], \quad M = \rho B, \quad C = (1 - \rho)B.$$

Operational implementations may include:

- Retrieval gating frequency
- Cache size or summary retention limits
- Confidence-based memory stabilization thresholds
- Token limits coupled to memory allocation
- LRU eviction thresholds

The protocol does not prescribe a specific architecture; it requires only that allocation between preserved structure and active recomputation be experimentally controllable.

4 Task Classes Suitable for Evaluation

The protocol recommends tasks that expose recursive re-entry and make instability measurable:

- Recursive summarization (multi-pass compression and query)
- Plan–critique–revise loops
- Retrieval vs recomputation under constrained context
- Multi-step reasoning requiring intermediate state preservation

Tasks should:

- Allow repeated recursion depth T
- Permit measurable intermediate outputs
- Have stable reference answers or scoring functions

5 Metrics

The protocol defines three primary observables.

5.1 Recomputation Churn (RC)

Measures repeated rediscovery of structure:

$$\mathrm{RC} = \frac{\text{Number of repeated derivations}}{\text{Total derivations}}.$$

Repeated derivations may be detected via:

- Exact string matching
- Embedding similarity threshold
- Tool-call duplication
- Identical intermediate claim reconstruction

5.2 Semantic Drift (DS)

Measures deviation across recursive steps:

$$\mathrm{DS}_t = 1 - \mathrm{sim}(y_t, y^\star),$$

where sim may be:

- Exact match
- F1 overlap
- Entailment score
- Embedding similarity

5.3 Utility Under Constraint (U)

Utility may be defined as:

$$U = \mathrm{Accuracy} - \alpha\mathrm{RC} - \beta\mathrm{DS},$$

or reported as separate curves for transparency.

6 Experimental Procedure

For each task instance:

1. Fix total resource budget B.
2. Select grid $\rho \in \{0.05, 0.10, \ldots, 0.95\}$.
3. Execute recursion depth T (e.g., $T = 5$–10).

4. Record $U(\rho)$, $\mathrm{RC}(\rho)$, $\mathrm{DS}(\rho)$.
5. Aggregate across task instances.
6. Plot curves vs ρ.

Confidence intervals should be reported where applicable.

7 Expected Footprint (Theoretical)

The Razor predicts:

- High recomputation churn in compute-heavy regimes.
- Reduced drift and churn in intermediate regimes.
- Rigidity or under-expression in memory-heavy regimes.
- A peaked utility curve with interior maximum.

This paper does not claim that such a footprint has been observed. It specifies the conditions under which it can be tested.

8 Scope and Limitations

- Results will depend on task family.
- Similarity metrics influence drift measurement.
- Allocation control mechanisms vary by architecture.
- This protocol does not prescribe model design.

The objective is reproducibility, not normative enforcement.

9 Conclusion

Preprints v1.1 and v1.2 formalized Robbie's Razor as a structural stability constraint under finite resources. This paper provides a reproducible empirical validation protocol for testing the stability-minimum hypothesis under controlled allocation.

Future work will report measured stability-footprint curves using this protocol.

Code and artifacts

Benchmark surface:

`https://github.com/RobbieRazor/robbies-razor-benchmarks`

References

J. Rissanen. Modeling by shortest data description. *Automatica*, 1978.

K. G. Wilson. The renormalization group. *Reviews of Modern Physics*, 1975.