Compress, Divide, Renew: Agent Architecture as Memory Lifecycles

Mark Lubin
mark@synix.dev

Abstract

Current approaches to agent memory treat it as an auxiliary retrieval problem—a database bolted onto an otherwise memoryless reasoning engine. We argue this framing is fundamentally inadequate. Working in a Gaussian multi-scale source model under squared-error distortion, we show that a bounded-resource processor operating over time is strictly better off satisfying three structural properties: it should *compress* lossily within bounded state, *partition* its state across timescales, and periodically *renew* each partition. Partitioning is not merely more efficient but more robust: a monolithic system's worst-case perturbation vulnerability scales linearly with the number of timescale components, whereas a partitioned system confines damage to a single component. Under an additional assumption (Compression Transparency), all three properties recur at every level of the resulting hierarchy. Prior work derives hierarchy from information compression or stability-plasticity tradeoffs; none derives renewal as a property at every level. We conjecture the framework extends beyond the Gaussian setting and state falsifiable predictions grounded against current benchmark evidence.

1 Setup: The Persistent Processing Problem

We formalize the persistent processing problem below. Section 2 provides a plain-language overview of all results before the formal development begins.

1.1 The System

Definition 1.1 (Persistent Information Processor)**.** *A* persistent information processor *is a tuple $\mathcal{S} = (C, B, c_{\text{ext}})$ operating in discrete time $t = 0, 1, 2, \ldots$: the system maintains an internal state $\sigma_t \in \{0,1\}^C$ of at most C bits, executes at most B unit-cost operations per step, and may access an external store of unbounded capacity at a cost of $c_{\text{ext}} \geq 1$ operations per access from the per-step budget B. At each step the system receives an observation $x_t \in \mathcal{X}$ and computes the update $\sigma_{t+1} = f_t(\sigma_t, x_t)$ within the budget B.*

The effective per-step accessible state is bounded by $C_{\text{eff}} = C + \lfloor B/c_{\text{ext}} \rfloor \cdot w$, where w is the word size of each external access. All subsequent references to "bounded state" concern C_{eff}.

1.2 The Environment

Definition 1.2 (Task-Relevant Information Source)**.** *The environment produces an observation sequence $(x_t)_{t \geq 0}$ drawn from a stochastic process on $\mathcal{X}$, with task relevance defined relative to a* query family *$\mathcal{Q}$ whose ground-truth answers $a_t(q)$ may vary with time. The* task-relevant entropy rate *is*

$$h \;=\; \lim_{T \to \infty} \frac{1}{T}\, H\big(A_T(Q) \,\big|\, A_0(Q)\big), \tag{1}$$

where $A_t(Q) = \{a_t(q) : q \in \mathcal{Q}\}$. An environment is non-trivial *if $h > 0$: new task-relevant information arrives indefinitely.*

Definition 1.3 (Multi-Scale Source Decomposition)**.** *The source admits a K-scale decomposition ($K \geq 2$) if there exist component processes $S_1, \ldots, S_K$ with (i) additivity: $x_t = g(S_1(t), \ldots, S_K(t))$; (ii) approximate independence: $I(S_i; S_j) \leq \mu$ for all $i \neq j$; (iii) ordered, well-separated timescales $\tau_1 < \cdots < \tau_K$ with $\tau_{i+1}/\tau_i \geq r_{\min} > 1$, where $\tau_i = \int_0^\infty |\rho_i(s)|\, ds$ is the integrated autocorrelation time; and (iv) non-degeneracy: each component contributes at least $h_{\min} > 0$ ongoing entropy rate. See Appendix A for the full formal statement.*

1.3 Performance Requirements

Definition 1.4 (Coherence and Bounded Response)**.** *The system $\mathcal{S}$ is* (ε, δ)-coherent *with respect to $\mathcal{Q}$ at time T if $\Pr[\, d(\hat{a}_T(q),\, a_T(q)) \leq \varepsilon\,] \geq 1 - \delta$ for a query q drawn uniformly from $\mathcal{Q}$, where d is a task-appropriate distortion measure. The system satisfies the* bounded response requirement *if there exist constants B_q and L_q, independent of elapsed time T, such that every query $q \in \mathcal{Q}$ is answered using at most B_q operations and L_q bits of output. The threshold ε reappears in Theorem 5.3 to distinguish regimes where renewal is needed from those where persistence suffices.*

1.4 Non-Stationarity: The Drift Axiom

Axiom 1 (Drift)**.** *For each timescale component $i \in \{1, \ldots, K\}$, the distribution $P_i(t)$ of task-relevant information produced by S_i is non-stationary over horizons longer than τ_i. Formally, the cumulative distributional divergence is unbounded:*

$$D_{\mathrm{KL}}\big(P_i(t) \,\|\, P_i(0)\big) \;\to\; \infty \quad \text{as } t \to \infty. \tag{2}$$

See Appendix A for the full formalization. The Drift Axiom is a *sufficient* condition for renewal: it guarantees unbounded degradation growth and hence finite τ^* (Theorem 5.3(a)). A drift-independent justification via perturbation accumulation is developed in Section 5.

Formal verification. Several algebraic identities in the bridge lemma (Lemma 5.1) have been verified in Lean 4 with Mathlib; the probabilistic arguments remain pen-and-paper proofs (Appendix J).

2 Overview of Results

Before the formal development, we sketch the three problems and their consequences in plain language.

Problem 1: Compression is lossy and permanent. Any system with finite memory that receives a continuous stream of new information must eventually forget something. This is unavoidable: once the incoming information exceeds what the memory can hold, some of it is discarded. The deeper problem is that this loss is *permanent*. No amount of clever internal reorganization can recover discarded information—it can only be restored if the environment independently provides it again. Moreover, the set of questions the system can no longer answer grows monotonically over time. This is not a design flaw; it is a mathematical consequence of finite capacity meeting unbounded input (Section 3).

Problem 2: Division into specialized components helps. Given that compression is unavoidable, the question becomes how to organize it. A system tracking a world that contains both fast-changing and slow-changing phenomena faces a dilemma: a single update rate either misses fast changes or wastes resources re-encoding slow ones that have not changed. Partitioning the system's state into components—each matched to a different timescale—strictly

reduces worst-case error. This is the familiar intuition behind separating working memory from long-term memory, but the formal result also shows a robustness benefit: a perturbation (corruption, noise, adversarial interference) to a partitioned system damages only one component, while the same perturbation to a monolithic system can degrade information at every timescale simultaneously (Section 4).

Problem 3: Periodic renewal is cost-optimal. Even after partitioning, each component's internal representation degrades over time. As the world drifts, the component's encoding scheme—optimized for the world as it was—becomes progressively mismatched to the world as it is. The system attends to features that no longer matter and ignores features that have become important. Independently, stochastic perturbations accumulate irreversibly. Eventually, rebuilding a component from scratch costs less than continuing to operate the degraded one. This yields an optimal *lifespan* for each component: replace when the instantaneous degradation equals the long-run average cost (Section 5).

Scale invariance: the recursion. These three problems do not arise once and then stop. Each component produces compressed output that becomes the input for the next level of the hierarchy. Under a transparency condition (the compressed output still carries genuine information about longer-timescale structure), the same three problems—compression, division, renewal—recur at every level. The hierarchy's depth is set by the source: it extends until no further timescale structure remains or coordination costs exceed the benefit. The result is a multi-level architecture that emerges from information constraints, not from engineering convention (Section 6).

What this means for practitioners. The framework predicts that agent memory systems should exhibit three properties: (1) specialized components operating at different timescales, (2) lossy compression within each component with monotonically growing blind spots, and (3) periodic rebuilding of each component. Current systems implement (1) partially and (2) implicitly; none implements (3). We state falsifiable predictions (Section 7.3) including a resource–architecture crossover time beyond which a well-structured system on modest resources outperforms a monolithic system on much larger resources.

3 Problem 1: Compression Is Lossy and Permanent

We now formalize the compression result sketched in Section 2. The result applies to any bounded-state processor; once partitions are established (Section 4), it applies to each partition individually.

3.1 Setup and Definitions

Consider a processor with state capacity C bits, receiving observations from a source with entropy rate $h > 0$ relative to query family $\mathcal{Q}$. Let $\mathcal{Q}_t^* \subseteq \mathcal{Q}$ denote the set of queries answerable by an oracle that records all observations verbatim up to time t, and let $\eta(q) > 0$ be the minimum mutual information needed to answer q reliably. The *unanswerable set* $U_t = \{q \in \mathcal{Q}_t^* : I(A_q; \sigma_t) < \eta(q)\}$ collects queries answerable by the oracle but not by the processor's bounded state. The state-update rule is:

$$\sigma_{t+1} \;=\; f_t\big(\sigma_t,\, X_{t+1}\big), \tag{3}$$

where $f_t : \{0,1\}^C \times \mathcal{X} \to \{0,1\}^C$.

3.2 Main Theorem

Theorem 3.1 (Irreversible Compression)**.** *Let a bounded-state processor have state capacity C bits and receive observations from a source with entropy rate $h > 0$ relative to query family $\mathcal{Q}$.*

(a) ***(Existence of loss.)*** *There exists a finite threshold $T^* \leq C/h$ such that for all $T > T^*$, $U_T \neq \varnothing$.*

(b) ***(Permanence.)*** *For any $q \in U_t$ and $t' \geq t$, we have $q \in U_{t'}$, provided no new observation independently provides the information needed for q.*

(c) ***(Monotonicity.)*** *If the source is non-redundant with respect to $\mathcal{Q}$, then $t' \geq t \Longrightarrow U_t \subseteq U_{t'}$.*

Proof sketch. **(a) Pigeonhole.** The cumulative task-relevant entropy grows at rate h. After $T^* = C/h$ steps, $H(X_1, \ldots, X_T \mid \mathcal{Q}) > C$, so the state $\sigma_T \in \{0,1\}^C$ cannot represent the full history injectively. By Fano's inequality, at least one query in $\mathcal{Q}_T^*$ has $I(A_q; \sigma_T) < \eta(q)$, giving $U_T \neq \varnothing$.

(b) Data Processing Inequality. For any past observation X_j with $j \leq t$, the chain $X_j \to \sigma_t \to \sigma_{t'}$ is Markov (since $\sigma_{t+1} = f_t(\sigma_t, X_{t+1})$ and X_{t+1} is conditionally independent of X_j given σ_t). By the DPI, $I(X_j; \sigma_{t'}) \leq I(X_j; \sigma_t)$. Once $I(A_q; \sigma_t) < \eta(q)$, no internal processing can recover the lost information: the loss is permanent.

(c) Non-redundancy. Under non-redundancy, future observations cannot compensate for information discarded from past observations. The information about A_q in $\sigma_{t'}$ decomposes into a component from σ_t (non-increasing by DPI) and a component from new observations (insufficient by non-redundancy). Hence $q \in U_t$ implies $q \in U_{t'}$.

Full proof in Appendix C. □

3.3 Rate of Information Loss

Corollary 3.2 (Growth rate of the unanswerable set)**.** *Under the conditions of Theorem 3.1, the information deficit grows at least linearly for $T > T^*$:*

$$H(X_1, \ldots, X_T \mid \mathcal{Q}) - I(X_1, \ldots, X_T; \sigma_T) \;\geq\; h \cdot T - C.$$

4 Problem 2: Division

Given that compression is inevitable (Theorem 3.1), the question is how to organize it. We formalize the partition advantage sketched in Section 2.

4.1 Source Model

We model each component as a stationary OU/AR(1) process with timescale τ_i, stationary variance σ_i^2, and entropy rate h_i. The distortion measure is squared error.

4.2 Temporal Matching

By classical water-filling [Berger, 1971, Cover and Thomas, 2006], optimal rate allocation decomposes additively across independent components; successive refinement is optimal [Equitz and Cover, 1991] (Appendix B).

Beyond rate allocation, matching the temporal update rate to each component's timescale provides a strict advantage. The key mechanism is *temporal aliasing*: an encoder sampling at rate r can track spectral content only up to the Nyquist frequency $r/2$.

Theorem 4.1 (Temporal Matching — Aliasing Penalty)**.** *Let $X(t) = X_1(t) + X_2(t)$ be the sum of two independent Gaussian AR(1) processes with characteristic timescales $\tau_1 < \tau_2$ and stationary variances σ_1^2, σ_2^2. Let the timescale separation ratio be $\alpha = \tau_2/\tau_1 \geq \alpha_{\min} > 1$.*

Consider two encoding schemes, both with total state capacity C bits and total encoding rate R bits per unit time: a homogeneous encoder $\mathcal{E}^{\text{hom}}(C, r)$ *sampling at a single rate r, and a* matched-rate partition $\{\mathcal{E}_1(C_1, r_1), \mathcal{E}_2(C_2, r_2)\}$ *with $C_1 + C_2 = C$ and $r_i = 1/\tau_i$.*

Then:

$$\min_{r>0} \max_{i\in\{1,2\}} D_i^{\text{hom}}(C, r) \;>\; \max_{i\in\{1,2\}} D_i^{\text{part}}(C_i, r_i), \tag{4}$$

where $\{C_1, C_2\}$ is the optimal capacity allocation and both sides use optimal rate allocation across components. The gap is bounded below by $\Delta_{\text{alias}} \geq \sigma_{\min}^2/(3\alpha)$. (This lower bound is loose: it reflects the minimax crossover point, where the monolithic encoder's best compromise rate falls between the two timescales. The actual advantage grows with separation because the fast-component aliasing worsens; a tighter bound is derived in Appendix B.)

Proof sketch. The homogeneous encoder faces a sampling dilemma: at any single rate r, it either undersamples fast components (incurring interpolation loss from Lemma B.1 in Appendix B) or oversamples slow ones (wasting capacity on redundant updates). The matched-rate partition eliminates this tradeoff. The gap arises from the minimax crossover of the two components' interpolation distortions. Full proof in Appendix B. □

Remark 4.1 (On General (Internally-Partitioning) Encoders)**.** *Theorem 4.1 restricts the monolithic encoder to the homogeneous class. A general encoder that internally maintains separate sub-states at different update rates can simulate any matched-rate partition — but this is not a counterexample; it* is *partitioning, embedded in a single system. The theorem establishes that the* encoding strategy *must partition state and update rates across timescales, whether or not the system is physically separated. The downstream argument (Sections 5–6) requires this* functional *partition regardless of system topology.*

Remark 4.2 (Beyond Timescale Separation)**.** *The partition-necessity argument does not fundamentally require components to differ in* timescale*. The core mechanism is that a single compression scheme incurs a quantified penalty when serving statistically heterogeneous components—timescale separation (Theorem 4.1) is one instance. The same structure appears whenever components differ in covariance eigenbasis, support geometry, or any property that makes a single rate-distortion codebook suboptimal for both. In particular,* horizontal scaling *may create this condition: multiple instances under load balancing are exposed to different distribution slices, potentially driving emergent specialization without centralized design. Formalizing this spatial axis of division requires extending the source model beyond the additive OU/AR(1) setting; we leave this as future work (Section 9).*

4.3 Robustness Implication: Fragility of Monolithic Encoders

The rate-allocation and temporal-matching results establish that partitioning is *more efficient* (lower distortion at equal resources). A complementary result shows that monolithic encoding is also *more fragile*: a single perturbation to a monolithic encoder can damage information at every timescale simultaneously, whereas a partitioned encoder confines damage to a single partition.

Definition 4.1 (Perturbation and Information Damage)**.** *A* perturbation *of magnitude ε at time t is a modification to the encoder state: $\sigma_t' = \sigma_t \oplus \delta_t$ (bitwise XOR), where $\|\delta_t\|_H \leq \varepsilon$ (Hamming distance on $\{0,1\}^C$). The* information damage *is $\Lambda(\delta, t) = \sum_{q\in\mathcal{Q}} \max\big(0,\; I(A_q; \sigma_t) - I(A_q; \sigma_t')\big)$, the total mutual information about query answers lost due to the perturbation.*

Corollary 4.2 (Perturbation Vulnerability of Monolithic Encoders)**.** *Let $X(t) = \sum_{i=1}^{K} X_i(t)$ be a K-component source as in Section 1. Consider a homogeneous encoder $\mathcal{E}^{\text{hom}}(C, r)$ and a matched-rate partition $\{\mathcal{E}_i(C_i, r_i)\}_{i=1}^{K}$ with $\sum_i C_i = C$. For any perturbation δ with $\|\delta\|_H = \varepsilon$:*

(a) ***Monolithic.*** *There exists a perturbation δ^* such that $\Lambda(\delta^*, t) \geq \varepsilon \cdot h_{\text{total}}$, where $h_{\text{total}} = \sum_i h_i$. The damage is distributed across all K timescale components.*

(b) ***Partitioned.*** *For any perturbation δ applied to partition j, the information damage satisfies $\Lambda(\delta, t) \leq \varepsilon \cdot h_j + O(\mu\,\varepsilon)$. The damage is confined to timescale τ_j.*

(c) ***Robustness ratio.*** $\Lambda_{\text{mono}}^{\max} / \Lambda_{\text{part}}^{\max} \geq K\,(1 - O(\mu / h_{\min}))$.

Proof sketch. Part (a): monolithic bits co-represent all components; adversarial perturbation targets any. Part (b): partitioned bits are isolated. Part (c): ratio follows from (a) and (b). Full proof in Appendix B. □

5 Problem 3 — Renewal

We formalize the renewal result sketched in Section 2. Two mechanisms drive degradation growth: **distributional drift** (the source shifts and a fixed encoder attends to obsolete features) and **perturbation accumulation** (stochastic damage is permanent by the DPI). The Drift Axiom is *sufficient* for renewal, not *necessary*.

5.1 Bridge Lemma: Fixed-Encoder Distortion Grows Under Drift

Under the Gaussian model of Section 4, the rate-R optimal encoder under $P_0 = \mathcal{N}(\mu_0, \Sigma_0)$ projects onto the top-k eigenvectors of Σ_0 via Π_0, discarding $\Pi_0^{\perp} = I - \Pi_0$.

Lemma 5.1 (Gaussian Fixed-Encoder Distortion Growth)**.** *With the encoder Π_0 held fixed (not re-optimized for P_t), the fixed-encoder distortion admits the exact decomposition:*

$$D(\Pi_0, P_t) \;=\; D^*(R, P_0) + \underbrace{\operatorname{tr}\!\left(\Pi_0^{\perp}\,(\Sigma_t - \Sigma_0)\,\Pi_0^{\perp}\right)}_{\text{variance growth in discarded subspace}} + \underbrace{\|\Pi_0^{\perp}\,(\mu_t - \mu_0)\|^2}_{\text{mean shift into discarded subspace}} + \Delta_Q, \tag{5}$$

where $\Delta_Q = D_Q(\Pi_0, P_t) - D_Q^*(R, P_0)$ is the quantization mismatch within the retained subspace (bounded: $|\Delta_Q| \leq M$ for a constant M depending on the codebook geometry).

Proof sketch. The fixed encoder subtracts μ_0, projects onto V_0 via Π_0, and quantizes. Under P_t, $X - \mu_0 \sim \mathcal{N}(\mu_t - \mu_0,\, \Sigma_t)$, so the projection loss picks up variance growth in the discarded subspace and mean shift into the discarded subspace. The quantization mismatch Δ_Q is bounded. Full proof in Appendix D. □

Corollary 5.2 (Distortion diverges under subspace drift)**.** *Under the Drift Axiom, the bridge lemma guarantees divergence provided drift manifests in the* discarded *subspace. We make this explicit:*

Assumption 1 (Subspace Drift)**.** *At least one of $\|\Pi_0^{\perp}(\mu_t - \mu_0)\| \to \infty$ or $\operatorname{tr}(\Pi_0^{\perp}\,\Sigma_t\,\Pi_0^{\perp}) \to \infty$.*

Subspace Drift is satisfied whenever drift has any component outside the retained principal subspace—the generic case, since Π_0 retains only $k < d$ dimensions. It fails only when all drift is confined to the retained subspace, in which case the fixed encoder remains well-matched and Theorem 5.3(c) applies: persistence is optimal.
Under Subspace Drift, $D(\Pi_0, P_t) \to \infty$. Under linear mean drift $\mu_t = \mu_0 + v\,t$ with $\Pi_0^{\perp} v \neq 0$:

$$D(\Pi_0, P_t) = D^*(R, P_0) + \|\Pi_0^{\perp} v\|^2\, t^2 + O(t). \tag{6}$$

Proof. By (5), $D(\Pi_0, P_t) \geq \|\Pi_0^{\perp}(\mu_t - \mu_0)\|^2 - M$, which diverges. The growth rate follows by substituting $\mu_t = \mu_0 + v\,t$ into (5). □

Table 1: Where the three problems operate in standard extensions.

Extension	**Compression layer**	**Degradation mode**
RAG	Index/query schema	Embedding drift: vectors miss new entities
Summarization	Summary schema	Tail loss: rare details dropped
Context expansion	Attention allocation	Dilution Liu et al. [2024]
Full recomputation	Retrieval selection	Selection criteria lag shift

5.2 Renewal Is Cost-Optimal

The following result is classical in reliability theory Barlow and Proschan [1965], Nakagawa [2005]; the contribution here is the bridge lemma above, which establishes that information-processing partitions satisfy the degradation conditions.

Define the long-run average cost of operating a partition with refresh interval τ:

$$J(\tau) \;=\; \frac{1}{\tau}\left[\int_0^\tau D(t)\,\mathrm{d}t \;+\; K\right], \tag{7}$$

where $D(t) \geq 0$ is the degradation at time t since last refresh, and $K > 0$ is the cost of refreshing.

Theorem 5.3 (Finite Optimal Lifespan)**.** *Let $D : [0, \infty) \to [0, \infty)$ be a continuous degradation function satisfying $D(0) = 0$, and let $K > 0$ be the replacement cost.*

(a) *If $D(t) \to \infty$ as $t \to \infty$ (drift-dependent or unbounded perturbation accumulation), then $J(\tau)$ achieves its minimum at a finite $\tau^* \in (0, \infty)$.*

(b) *If $D(t) \to D_{\max} < \infty$ and $D_{\max} > \varepsilon$ (the coherence threshold from Definition 1.4), then $J(\tau)$ is eventually decreasing but $D_{\max}$ exceeds the performance requirement: renewal is still needed to reset below ε, though the optimal τ^* may be longer.*

(c) *If $D(t) \to D_{\max} \leq \varepsilon$, then $\tau^* = \infty$: persistence is optimal and renewal is unnecessary.*

Proof sketch. **(a)** As $\tau \to 0^+$, $J(\tau) = K/\tau + o(1) \to \infty$ (replacement cost dominates). As $\tau \to \infty$, since $D(t) \to \infty$, $J(\tau) \to \infty$ (degradation dominates). By continuity and the extreme value theorem, J attains a finite minimum at some $\tau^* \in (0, \infty)$. The optimality condition is:

$$\boxed{D(\tau^*) \;=\; J(\tau^*)} \tag{8}$$

Replace when the instantaneous degradation equals the long-run average cost. **(b)** When $D_{\max} > \varepsilon$, the system eventually violates coherence; renewal resets D to zero. **(c)** When $D_{\max} \leq \varepsilon$, $J(\tau)$ is monotonically decreasing for large τ, so $\tau^* = \infty$. Full proof in Appendix D. □

When degradation follows a power law $D(t) = \alpha t^\beta$, the optimal lifespan has a closed form: $\tau^* = ((\beta + 1)K/(\alpha\beta))^{1/(\beta+1)}$ (Appendix D).

Remark 5.1 (Extensions relocate, not eliminate)**.** *Retrieval augmentation, context expansion, and external storage do not circumvent the bounded-state limitations driving Theorems 3.1–5.3. Any extension mechanism with bounded local state, bounded per-step compute, and online operation under drift faces Compression, Division, and Renewal at its own operational layer (Table 1).*

6 Scale Invariance — The Recursion

We formalize the recursive argument sketched in Section 2.

6.1 The Recursive Argument

The argument proceeds in four steps.

(a) Compressed output is input to the next level. Partition i at timescale τ_i produces a compressed representation that constitutes the input partition $i+1$ must process at τ_{i+1}.

(b) The compressed stream has nonzero entropy. For the recursion to proceed, the compressed output of level i must carry genuine information about structure at τ_{i+1}.

Assumption 2 (Compression Transparency)**.** *Let $\{X_{\tau_1}, \ldots, X_{\tau_K}\}$ be the timescale decomposition of the source (Section 1), and let $\hat{X}_{\tau_i}$ denote the minimum-distortion compression output at level i under capacity C_i. For each level $i \in \{0, \ldots, K-2\}$,*

$$H\Big(X_{\tau_{i+1}} \mid \hat{X}_{\tau_i}\Big) \;<\; H\big(X_{\tau_{i+1}}\big) - h_{\min}, \tag{9}$$

where $h_{\min} > 0$ is a uniform lower bound on the preserved mutual information. Equivalently,

$$I\Big(X_{\tau_{i+1}}; \hat{X}_{\tau_i}\Big) \;\geq\; h_{\min} \;>\; 0. \tag{10}$$

(c) Same constraints, same problems. If Compression Transparency holds at level i, then level $i+1$ receives input with entropy rate $\geq h_{\min} > 0$, under bounded capacity and drift. Theorems 3.1–5.3 apply.

(d) Termination. The recursion halts when $\tau_k \geq T$, $h_k \to 0$, transparency fails, or coordination cost exceeds marginal benefit.

6.2 Formal Statement and Proof

Theorem 6.1 (Recursive Structure Under Separable Sources)**.** *Let $\mathcal{S}$ be a source with nontrivial structure at K timescales $\tau_1 < \cdots < \tau_K$ satisfying the Drift Axiom (Section 1) and Compression Transparency (Assumption 2). Let $\mathcal{M}$ be any bounded-state system with total capacity C maintaining bounded distortion $D_i \leq D_{\max}$ at each τ_i. Then $\mathcal{M}$ achieves strictly lower worst-case distortion by implementing:*

(a) At least K functional partitions with update rates matched to their respective timescales (Theorem 4.1).

(b) Lossy compression within each partition, with permanent, non-decreasing information loss (Theorem 3.1).

(c) A finite refresh interval $\tau_i^* \in (0, \infty)$ for each partition under unbounded drift (Theorem 5.3).

(d) Information flow from faster to slower partitions: the input to partition $i+1$ is the compressed output of partition i.

The hierarchy depth is bounded:

$$K \;\leq\; \frac{\log(T/\tau_{\min})}{\log r} \tag{11}$$

for minimum source timescale $\tau_{\min} = \tau_1$ and timescale ratio $r = \inf_i \tau_{i+1}/\tau_i > 1$.

Proof sketch. By strong induction. *Base case:* at τ_1, the source delivers information at rate $h_1 > 0$ into bounded capacity C_1 under drift; Theorems 3.1–5.3 apply directly. *Inductive step:* Compression Transparency ensures the compressed output of level k has nonzero entropy rate about τ_{k+1}; the Drift Axiom propagates through compression; therefore Theorems 3.1–5.3 apply at level $k+1$. The recursion halts when source structure is exhausted. Depth bounded by $K \leq \log(T/\tau_{\min})/\log(r)$. Full proof in Appendix F. □

Coordination cost (synchronization overhead per added level) bounds the optimal depth to a finite K^*, typically 2–4 levels in practice.

7 Empirical Evidence and Predictions

The three problems are mathematical consequences of the system model. The evidence below is *consistent with* the framework, not *uniquely supporting*: no existing benchmark was designed to test the three problems directly.

7.1 Division

On BEAM Tavakoli et al. [2025], Qwen2.5-32B drops from 0.280 accuracy at 100K tokens to 0.133 at 10M tokens (52.5% degradation). The multi-component LIGHT system outperforms the vanilla baseline by 107–156% at scale. Hindsight's four epistemic networks achieve 83.6% on LongMemEval, outperforming flat baselines by a wide margin Latimer et al. [2025]. Multi-component architectures that separate information by scope are consistently more robust, consistent with Theorem 4.1.

7.2 Compression

On LoCoMo Maharana et al. [2024], the best model achieves 25.0% accuracy on temporal reasoning versus 92.6% human accuracy—within context windows averaging 9,209 tokens. LoCoMo's summarization evaluation yields FactScore F_1 of only 45.9%, directly demonstrating information loss under compression consistent with Theorem 3.1: bounded-state compression is lossy, and the loss manifests on queries the scheme was not optimized for.

7.3 Predictions

We state two sharp falsifiable predictions; three additional predictions (P1, P2, P4) are summarized in Table 2.

Prediction 1 (Resource–Architecture Crossover)**.** *There exists a computable time T_{cross} beyond which a partitioned-and-renewed system on modest resources outperforms a monolithic system on $R\times$ larger resources ($R \geq 10$). T_{cross} is determined by the degradation exponent, the resource ratio, and the renewal cost. A worked estimate (Appendix H) yields $T_{\text{cross}} \approx 3.1$ years for a personal assistant scenario with $10\times$ resource ratio.*

Disconfirmation criterion. No crossover observed within 12 months at a $10\times$ resource ratio, with the monolithic system maintaining $> 90\%$ of its initial advantage throughout.

Prediction 2 (Renewal Necessity)**.** *A partitioned hierarchical system with periodic renewal (components refreshed at intervals τ_i^* for each level i) outperforms an architecturally identical hierarchy without renewal. The performance gap grows with T. Specifically: the renewed system maintains surprise-query accuracy within a bounded envelope independent of T, while the persistent system's accuracy degrades as $\Omega(T^\alpha)$ for some $\alpha > 0$.*

Table 2: Falsifiable predictions. Novelty assessed relative to prior work on multi-scale processing, continual learning, and agent memory.

#	Name	Tests	Novelty	Key Disconfirmation
P1	Competence Narrowing	Thm 3.1	Low	Flat system $> 80\%$ accuracy at $T = 6$ mo
P2	Partition Advantage	Thm 4.1	Low	Monolithic matches partitioned at $T > 3$ mo
P3	Crossover	Thms 4.1+5.3	High	No crossover at 12 mo, $10\times$ ratio
P4	Hierarchy Depth	Thm 6.1	Medium	K^* uncorrelated with (h, T) across 5+ environments
P5	Renewal Necessity	Thm 5.3	High	Persistent matches renewed at $T > \tau^*$

Prediction P5 is the sharpest test of this paper's central novel claim. A disconfirming result—hierarchy without renewal performing comparably to hierarchy with renewal—would undermine Theorem 5.3.

8 Related Work

Prior work derives hierarchy from information compression Schmidhuber [1992], stability-plasticity tradeoffs Grossberg [1980], and temporal receptive fields Hasson et al. [2015]; none derives renewal as a structural property at every level. Table 3 summarizes existing agent memory systems.

Theoretical predecessors. Rate-distortion memory models Sims [2016], Hahn et al. [2022] establish capacity-distortion tradeoffs for single-timescale memory. CLS theory McClelland et al. [1995] derives a two-timescale hierarchy from neurobiological constraints; we generalize to K levels from information-theoretic principles. Multi-timescale architectures Schmidhuber [1992], Koutnik et al. [2014], Chung et al. [2017], active inference Friston et al. [2018], and bounded-rational decision-making Genewein et al. [2015] all derive hierarchy; none derives renewal as a property at every level.

Continual learning and catastrophic forgetting. Grossberg Grossberg [1980] identified the stability-plasticity dilemma; McCloskey and Cohen McCloskey and Cohen [1989] demonstrated it empirically as catastrophic interference. The three problems we identify are the stability-plasticity dilemma decomposed into its constituent information-theoretic mechanisms and extended with renewal. Corollary 4.2 formalizes the robustness dimension: the K-fold perturbation vulnerability gap between monolithic and partitioned encoders is the information-theoretic generalization of the catastrophic interference phenomenon. EWC's effectiveness Kirkpatrick et al. [2017] is explained by this lens: the Fisher-information penalty creates an implicit two-timescale partition (protected slow parameters, plastic fast parameters), achieving the isolation that monolithic networks lack. Knoblauch et al. Knoblauch et al. [2020] show that avoiding forgetting under bounded capacity is computationally intractable; we show that information loss is information-theoretically inevitable and irreversible.

Runtime systems. Generational garbage collectors Lieberman and Hewitt [1983], Ungar [1984] partition the heap by object lifetime (division); JIT compilers compress interpreted code into specialized machine code (compression); dynamic deoptimization Hölzle et al. [1992] and the

Table 3: Coverage of the three structural properties by existing agent memory systems.

System	Division	Compression	Renewal
Generative Agents Park et al. [2023]	Partial (2-level)	Yes (reflection)	No
MemGPT Packer et al. [2023]	Yes (3-tier)	Yes (paging)	No
RAG Lewis et al. [2020]	No	Yes (index)	No
Hindsight Latimer et al. [2025]	Yes (4 networks)	Implicit	No
AgeMem Yu et al. [2026]	Yes	Yes	Yes (RL-trained)
LightMem Fang et al. [2025]	Yes (3-stage)	Yes (entropy-aware)	Partial (sleep-time)
MemoryBank Zhong et al. [2023]	No	Implicit	Passive (forgetting curve)

C4 collector's per-generation compaction Tene et al. [2011] implement renewal. These systems independently converge on CDR structure.

Agent memory systems. Generative Agents Park et al. [2023] implement a two-level architecture (observations + reflections) without renewal. MemGPT Packer et al. [2023] provides a three-tier hierarchy with explicit paging but static management policies and no renewal. RAG Lewis et al. [2020] compresses at the index layer and faces all three problems there. Hindsight Latimer et al. [2025] organizes memory into four epistemic networks that correlate with timescale, achieving 83.6% on LongMemEval. AgeMem Yu et al. [2026] trains update and delete policies via RL, coming closest to implementing all three CDR properties. LightMem Fang et al. [2025] uses a three-stage pipeline with entropy-aware compression and sleep-time consolidation (partial renewal). MemoryBank Zhong et al. [2023] applies Ebbinghaus forgetting curves—passive decay rather than active renewal (Section 9).

Empirical degradation evidence. Rath Rath [2026] introduces a 12-dimension Agent State Integrity metric measuring behavioral degradation in long-running agents; the observed decay curves are consistent with the bridge lemma's predictions. Dongre et al. Dongre et al. [2025] show that drift can equilibrate ($D(t) \to D_{\max}$) and that reminder interventions restore performance—the closest empirical analogue to CDR's renewal mechanism. Xiong et al. Xiong et al. [2025] demonstrate that errors in stored experience compound over time, providing direct evidence for Theorem 3.1(b). Hu et al. Hu et al. [2025] survey 45+ authors' perspectives on agent memory and identify memory evolution as an under-studied frontier.

9 Discussion

Fragility and renewal. Corollary 4.2 shows that partitioning provides isolation; Theorem 3.1(b) shows that perturbation damage within a partition is permanent. Renewal resets accumulated damage. As argued in Section 5, this justification is independent of drift: division provides isolation, renewal provides self-healing.

Biological convergence. Multicellularity implements CDR at the cellular level: cells partition function by type, compress experience into epigenetic state, and renew via cell division with bounded lifespans Kirkwood [1977].

Spatial division. In horizontally scaled deployments, load balancing exposes instances to different distribution slices, potentially driving emergent specialization. Formalizing this spatial axis requires extending the source model beyond the additive OU/AR(1) setting (Remark 4.2).

Active renewal vs. passive forgetting. Time-based decay Zhong et al. [2023] addresses storage pressure but not distribution mismatch: it removes entries without re-materializing the compression scheme from current data. In resource-constrained systems, forgetting is mandatory but the question is *what* to forget. We propose *weighted renewal*: replacement priority reflects both temporal staleness and utility decay (divergence between the entry's encoded distribution and the current source, as measured by $D(t)$). This reduces to pure time decay only when utility is uniform across entries.

Scope. The theory establishes structural advantages, not unique optimality: topology, tier counts, and timescale ratios are engineering decisions constrained but not determined by the framework.

Acknowledgements. This paper was drafted with assistance from Claude Opus 4.6 (Anthropic) and GPT-5.3 Coex (OpenAI). All claims, errors, and editorial decisions remain the author's.

References

Richard E Barlow and Frank Proschan. *Mathematical Theory of Reliability*. Wiley, 1965.

Toby Berger. *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Prentice-Hall, 1971.

Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. In *International Conference on Learning Representations*, 2017.

Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

Atharva Dongre et al. Drift no more: Addressing distribution shift in LLM agent memory systems. *arXiv preprint*, 2025.

William H R Equitz and Thomas M Cover. Successive refinement of information. *IEEE Transactions on Information Theory*, 37(2):269–275, 1991.

Yubo Fang et al. LightMem: Efficient memory management for long-context LLM agents. *arXiv preprint*, 2025.

Karl J Friston, Richard Rosch, Thomas Parr, Cathy Price, and Howard Bowman. Deep temporal models and active inference. *Neuroscience & Biobehavioral Reviews*, 90:486–501, 2018.

Tim Genewein, Felix Leibfried, Jordi Grau-Moya, and Daniel A Braun. Bounded rationality, abstraction, and hierarchical decision-making: An information-theoretic optimality principle. In *Frontiers in Robotics and AI*, volume 2, 2015.

Stephen Grossberg. How does a brain build a cognitive code? *Psychological Review*, 87(1):1–51, 1980.

Michael Hahn, Richard Futrell, Roger Levy, and Edward Gibson. A resource-rational model of human processing of recursive linguistic structure. *Proceedings of the National Academy of Sciences*, 119(43), 2022.

Uri Hasson, Janice Chen, and Christopher J Honey. Hierarchical process memory: Memory as an integral component of information processing. *Trends in Cognitive Sciences*, 19(6): 304–313, 2015.

Urs Hölzle, Craig Chambers, and David Ungar. Debugging optimized code with dynamic deoptimization. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '92)*, pages 32–43. ACM, 1992.

Zekun Hu et al. Memory in the age of AI agents: A comprehensive survey. *arXiv preprint*, 2025.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Thomas B L Kirkwood. Evolution of ageing. *Nature*, 270:301–304, 1977.

Jeremias Knoblauch, Hisham Husain, and Tom Diethe. Optimal continual learning has perfect memory and is NP-hard. In *International Conference on Machine Learning*, pages 5327–5337, 2020.

Jan Koutnik, Klaus Greff, Faustino Gomez, and Jürgen Schmidhuber. A clockwork RNN. In *International Conference on Machine Learning*, pages 1863–1871, 2014.

James Latimer et al. Hindsight: Episodic memory for long-horizon agents. *arXiv preprint*, 2025.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, 2020.

Henry Lieberman and Carl Hewitt. A real-time garbage collector based on the lifetimes of objects. *Communications of the ACM*, 26(6):419–429, 1983.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the ACL*, 2024.

Adyasha Maharana et al. LoCoMo: Long-context multi-turn dialogue understanding with mixed-initiative conversations. *arXiv preprint*, 2024.

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3): 419–457, 1995.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989.

Toshio Nakagawa. *Maintenance Theory of Reliability*. Springer, 2005.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. MemGPT: Towards LLMs as operating systems. In *arXiv preprint arXiv:2310.08560*, 2023.

Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *ACM Symposium on User Interface Software and Technology*, 2023.

Nishant Rath. Agent drift: Quantifying behavioral degradation in long-running LLM agents. *arXiv preprint*, 2026.

Jürgen Schmidhuber. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242, 1992.

Chris R Sims. Rate-distortion theory and human perception. *Cognition*, 152:181–198, 2016.

Arash Tavakoli et al. BEAM: A benchmark for evaluating agent memory over long horizons. *arXiv preprint*, 2025.

Gil Tene, Balaji Iyengar, and Michael Wolf. C4: The continuously concurrent compacting collector. In *Proceedings of the International Symposium on Memory Management (ISMM '11)*, pages 79–88. ACM, 2011.

David Ungar. Generation scavenging: A non-disruptive high performance storage reclamation algorithm. In *Proceedings of the ACM SIGSOFT/SIGPLAN Software Engineering Symposium on Practical Software Development Environments*, pages 157–167. ACM, 1984.

Wenhan Xiong et al. The impact of memory management on long-horizon agent performance. *arXiv preprint*, 2025. Harvard University.

Yichen Yu et al. AgeMem: RL-trained memory management for long-horizon agents. *arXiv preprint*, 2026.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. *arXiv preprint arXiv:2305.10250*, 2023.

A Setup: Supplementary Definitions

Full multi-scale source decomposition. *[Full formal statement of Definition 1.3, including regularity conditions, spectral gap requirements, and the relationship between integrated autocorrelation time and spectral density.]*

Drift proof. *[Complete proof that multi-scale structure with ongoing entropy implies unbounded KL divergence at each partition's timescale.]*

Exclusions. *[Discussion of what the system model excludes: stationary environments ($h = 0$), systems with growing memory, single-timescale sources ($K = 1$), and systems without bounded response requirements.]*

B Division Proofs

Rate Allocation (Classical Water-Filling). *[Full proof via classical water-filling for independent Gaussian components. References: Berger (1971), Cover & Thomas (2006), Equitz & Cover (1991) for successive refinability.]*

Lemma B.1 (Sampling Distortion)**.** [Interpolation distortion for an AR(1) process sampled at rate r when $r < 1/\tau$: the distortion floor from undersampling.]

Proof of Theorem 4.1 (Temporal Matching). *[Full proof of the aliasing penalty. Constructs the minimax crossover between fast- and slow-component interpolation distortions and derives the $\sigma^2_{\min}/(3\alpha)$ lower bound on the gap.]*

Minimax criterion and successive refinement. *[Remarks on the choice of minimax vs. weighted-sum distortion and the role of successive refinement optimality for Gaussian sources.]*

C Compression Proofs

Full definitions. *[Complete definitions of oracle $\mathcal{O}_T$, answerable set, unanswerable set U_t, non-redundancy condition, and the information threshold $\eta(q)$.]*

Proof of Theorem 3.1 (Irreversible Compression). *[Full proof of parts (a)–(c). Part (a) via pigeonhole + Fano's inequality. Part (b) via Markov chain argument and DPI. Part (c) via non-redundancy decomposition.]*

D Renewal Proofs

Proof of Lemma 5.1 (Gaussian Fixed-Encoder Distortion Growth). *[Full derivation of the exact decomposition* (5)*. Orthogonal decomposition of distortion into projection loss and quantization loss. Boundedness of Δ_Q under assumption (A2).]*

Proof of Theorem 5.3 (Finite Optimal Lifespan). *[Full proof via extreme value theorem and first-order optimality condition. Derivation of the optimality condition $D(\tau^*) = J(\tau^*)$.]*

Power-law degradation. *[Closed-form solution $\tau^* = ((\beta+1)K/(\alpha\beta))^{1/(\beta+1)}$ when $D(t) = \alpha t^\beta$. Comparative statics in α, β, K.]*

Bounded degradation. *[Proof that when $D(t) \to D_{\max} < \infty$, $\tau^* = \infty$ and persistence is optimal. Characterizes the boundary between renewal-required and renewal-unnecessary regimes.]*

E Extension Analysis

[Detailed analysis of how each extension mechanism (RAG, summarization, context expansion, full recomputation, multi-schema ensemble) satisfies the conditions of Theorems 3.1–5.3. Includes per-extension proofs that the bounded-state and bounded-compute conditions hold at the extension's operational layer.]

F Recursion Proofs

Proof of Theorem 6.1 (Recursive Necessity). *[Full inductive proof. Base case: direct application of Theorems 3.1–5.3 at τ_1. Inductive step: Compression Transparency propagation, Drift Axiom propagation through compression (KL minus bounded correction), re-application of Theorems 3.1–5.3. Termination conditions.]*

Compression Transparency analysis. *[When Assumption 2 holds and when it fails. Sufficient conditions in terms of source spectral structure and compression rate. Connection to successive refinability.]*

Timescale geometry. *[Proposition on geometric spacing of timescales. Proof that $\tau_i = \tau_1 \cdot r^{i-1}$ is optimal under uniform compression loss per level.]*

Coordination cost and finite optimal depth. *[Definition of total system cost including per-level coordination overhead. Proof that optimal depth K^* is finite. Typical values $K^* \in \{2, 3, 4\}$ under realistic parameters.]*

G Empirical Details

[Demand Scaling assumption. Detailed benchmark methodology notes. Extended numerical instantiation with sensitivity analysis.]

H Prediction Derivations

[Full derivations for the crossover time T_{cross} and renewal interval τ^* estimates in Predictions P3 and P5. Parameter sensitivity analysis.]

I Extended Related Work

[Extended discussion of theoretical predecessors, continual learning literature, agent memory systems, and connections to active inference / free energy minimization frameworks.]

J Formal Verification

Several supporting algebraic identities were verified in Lean 4 (v4.29.0-rc1) with Mathlib. All 11 formalized lemmas compile without `sorry` or axiom abuse. The verification covers deterministic algebraic steps only, not the probabilistic core of the proofs:

- **Interpolation distortion** (supporting Theorem 4.1): the OU bridge variance identity, the exponential-to-hyperbolic ratio conversion, and the key positivity inequality $\coth(x) > 1/x$ for all $x > 0$, proved via a monotonicity argument on the derivative $g'(t) = t(e^t - e^{-t}) > 0$.
- **Fixed-encoder distortion decomposition** (Lemma 5.1): trace linearity for the variance growth term and the norm-squared expansion for the mean shift term.
- **Drift growth** (Corollary 5.2): the quadratic growth identity $\|tv\|^2 = t^2\|v\|^2$ under linear mean drift.

Probabilistic arguments (conditional expectation, Gaussian regression, quantization mismatch bounds) were not formalized; these require probability-theory infrastructure beyond current Mathlib scope. No mathematical errors were found. Source files are available in the supplementary materials.