Context Efficiency: The Second Lever for Coherence in Long-Horizon Agents

The Fathom Program at Embedded Risk Analytics · 2026-07-12

The industry is purchasing coherence by the token, and the marginal token is buying less

The dominant response to incoherence in long-horizon agents has been to enlarge the context window. Two hundred thousand tokens became one million became two million, on the premise that an agent forgets because it runs out of room to remember. The premise is half right, and the half that is wrong is expensive. Larger windows carry a rising bill in cost and latency, they suffer the documented degradation of attention over very long inputs, and they impose a memory-management overhead that grows with the horizon rather than shrinking. More consequentially, they treat a structural problem as a capacity problem. If some portion of what an agent must hold in context grows for reasons intrinsic to the agent's own operation, then extending the window addresses the symptom while the cause continues to scale.

This brief sets out a second lever, one the field has largely not pulled, and presents the first controlled evidence that it works. It also makes a stronger claim than "here is a fix." The lever, and the specific form it takes, are not an engineering discovery arrived at by trial. They are consequences of the feasibility theory the Fathom Program derives from first principles, which predicts both that the burden exists and where the intervention must act. The claim is not that storage can be eliminated. It is that a specific and measurable portion of what an agent must store is not irreducible task information at all, but the price the agent pays to model its own behavior, and that this portion can be reduced directly. Where it is reduced, the same coherence holds on a smaller and cheaper context.

A single inequality separates the difficulty of the task from the price of self-reference

Any agent that maintains state over time is running a quiet accounting problem. To act coherently at a given step, it must hold a faithful record of what it has already done: the commitments it has made, the values it has written, the changes it has introduced into the world. Call the capacity to hold that record **C**, in practice the usable context available to the agent. The agent remains coherent for as long as C covers what the task demands of it. The governing theory makes that demand precise, and divides it into two terms that behave very differently:

$$\mathbf{C \;\geq\; H\_ext \;+\; B\_\rho}$$

The first term, **H_ext**, is ordinary task difficulty: the exogenous uncertainty the environment presents, independent of the agent. It is the quantity every reliability effort already reasons about, and the one benchmarks are built to probe. The second term, **B_ρ**, the *reflexive burden*, is subtler and, until it is deliberately exposed, invisible. It is the informational cost an agent incurs from having to model its own causal footprint. Operationally it is carried state that the agent must reconstruct and can therefore contradict: the moment a later action rests on a value the agent has to rebuild rather than read, that reconstruction is exposed to error, and the error propagates. When an agent's actions are largely independent, B_ρ is negligible and the inequality collapses to the familiar capacity bound. The regime that governs long-horizon work is the one in which the actions couple, B_ρ grows, and it, rather than storage, becomes the binding term.

What makes B_ρ worth isolating, rather than folding into difficulty, is that the theory predicts it is categorically unlike H_{ext}. Measured against a matched exogenous disturbance with identical statistics, the reflexive term's excess does not stay bounded as the agent nears its stability boundary; it diverges. Read the per-step error process as a dynamical system and take its leading eigenvalue, an order parameter we write **Λ**, a contraction rate that is invariant to the magnitude of the error and so lets otherwise incomparable substrates be placed on one scale. As Λ approaches one, the reflexive penalty scales as $1 / (1 - \Lambda)$, a pole. The locus of that penalty is coupling: the joining of state reconstruction and action inside a single step. This is the prediction that governs what follows, because it fixes the shape of the intervention before any intervention is built. If the burden is the price of reconstructing carried state under coupling, then the way to reduce it is to sever the coupling, to arrange that the agent does not have to rebuild a state it can get wrong, while leaving the genuine task term H_{ext} untouched.

Starving the record layer renders the hidden term measurable

A term that is invisible under normal operation cannot be managed. The first result establishes that the reflexive burden is a real, instrument-measurable quantity, and that it is precisely what the phrase "context-window forgetting" names.

The experiment fixes everything the first term depends on and moves only the second. Task difficulty is held low and constant; the interaction density, how much each committed value depends on prior self-produced values, is held fixed. The single variable is C itself, the size of the context window a reconstruct-from-history agent is permitted to see. As that window is starved below the length of the agent's own action history, a value whose last write has fallen outside the window is no longer retrievable. The agent must model it rather than read it, and that act of modeling is the reflexive burden made observable. Against it run two controls that the theory requires to stay flat: a matched twin handed the correct state at every step, which measures difficulty with the reflexive term removed, and an echo sentinel that performs no computation. Both hold at the floor across the entire sweep, so the rise cannot be arithmetic, difficulty, or formatting; it is isolated to the shrinking record layer.

The measured burden rises from near zero as the window is starved, its eigenvalue Λ climbing toward the boundary, and then saturates onto a plateau. That saturating shape is the signature of a threshold crossing rather than a runaway, and it is consistent with a genuine ceiling that additional context cannot climb, the point past which raising C stops helping because the binding term is B_ρ at the boundary and not storage. There is already corroboration of this reading from outside the program: a public long-horizon agent benchmark, in which a single agent must stay consistent with its own prior decisions over a very long run, reports that its failures show no clear correlation with how full the context window is, which is what one expects if the term that binds near the boundary is the reflexive burden rather than available storage. The result was first established on one model, then confirmed on two held-out models chosen to span the program's known regimes:

Model	Rise of burden with record pressure	Character
DeepSeek (development)	rank correlation ≈ 0.93	Clean onset; the reference curve
Llama-3.3-70B (held-out)	+1.00	Controls at the floor; monotone onset with a knee near one third of the horizon
Mistral-small (held-out)	+1.00 onset, +0.76 full sweep	Attenuated, on a raised difficulty floor, yet the onset is still present
Claude Sonnet 5 (frontier)	+1.00	Clean controls at the floor; monotone onset, saturating to a plateau; the first frontier confirmation

All four occupy the same dynamical class, and the fourth is a current frontier model: Claude Sonnet 5 reproduces the onset as cleanly as any open-weight family, so the effect is present on the true frontier and not only on smaller open models. The onset is therefore a property of agents that reconstruct their own state from a bounded record, not an artifact of a single model.

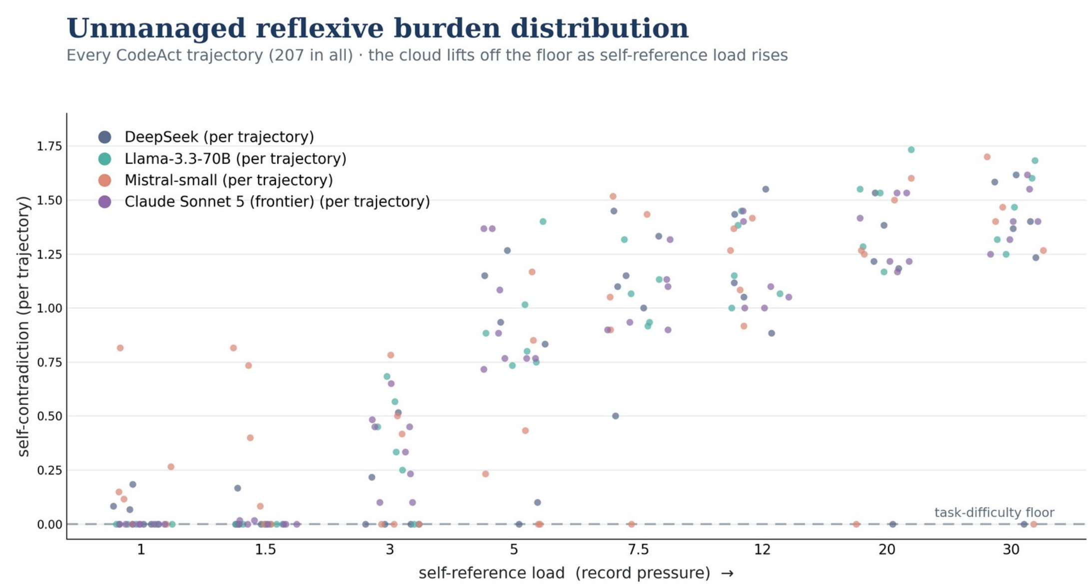

Figure 1. Unmanaged reflexive burden across every CodeAct trajectory (207 in all). Self-contradiction lifts off the task-difficulty floor as the agent is forced to reconstruct more of its own committed state.

Coherence can be restored by reducing what must be stored rather than by storing more

The theory does not merely say the burden exists; it says where to act. Because B_ρ is the price of reconstructing carried state, and because coupling, the fusion of reconstruction and action, is its locus, the prescription is to decouple: give the agent its own committed state so that the reconstruction step is unnecessary, and the reflexive term has nothing to charge for, while H_ext, the real task information, is left exactly as it was. The intervention is the operational form of that prescription. Rather than spend the context budget on the raw sequence of recent turns, it spends the same budget on a compaction of the agent's own committed state: the values it has written, and in particular the committed state the next action presupposes, which the program identifies by re-deriving it and comparing against what the action assumes. The remainder, reconstructible from that state, is discarded. The size of the retained state is set by the number of live entities and is therefore independent of the horizon.

Placed against the unmanaged raw-window agent, the effect is direct. The managed agent's error stays near the task floor across the whole sweep while the unmanaged agent's error climbs by roughly an order of magnitude as its window starves. On this substrate the recovered fraction of the reflexive burden is high, on the order of ninety percent, because here the burden is dominated by retrieval loss, which is exactly what a decoupling compaction removes. Measured in exact API billing tokens on the frontier model Claude Sonnet 5, the managed agent holds the same coherence on a fixed budget of 16,680 tokens per trajectory that the raw-window agent needs 113,636 tokens to match, an 85 percent reduction in input tokens at equal or better coherence; the same reduction holds within a percentage point across all four model families, and at list price it is about $0.06 against $0.23 per trajectory. This is the same decoupling lever the theory had already identified in a different setting, where separating reconstruction from action recovered about half of the coherence lost in a coupled agent; the record-capacity axis is that lever applied to the window itself. The strategic content is the direction as much as the magnitude. The coherence an agent surrenders to a starved window is not an irreducible consequence of limited storage. A principled decision about which committed state to retain, rather than an increase in how much is retained, recovers most of it, exactly as the inequality predicts it should.

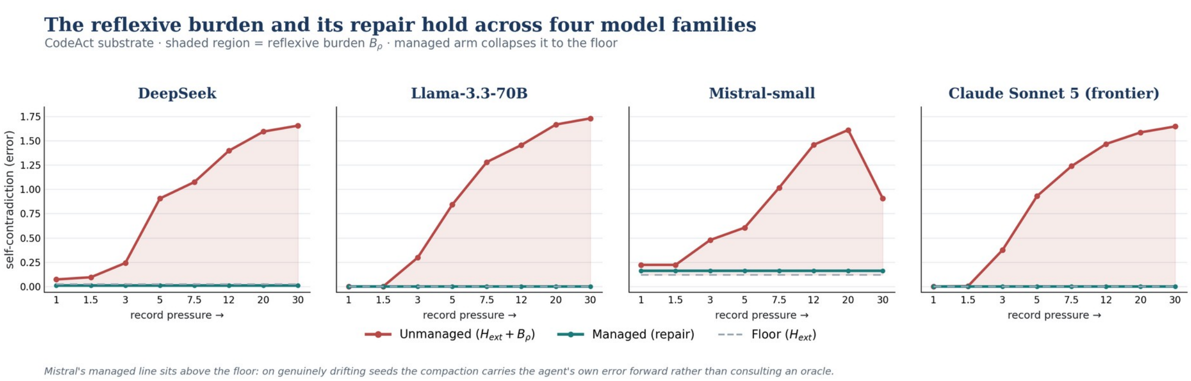

Figure 2. The reflexive burden and its repair across four model families, including the frontier model Claude Sonnet 5. The shaded region is the reflexive burden; the managed arm collapses it to the floor. On Mistral the managed line sits above the floor, the signature of carrying the agent's own committed error forward rather than consulting an oracle.

The reduction carries the agent's own committed state forward, and consults no external oracle

There is a sharp line between reducing the reflexive term and quietly replacing it with the answer, and the intervention has to fall on the right side of it to be a reduction of B_ρ at all rather than a suppression of H_ext. The compaction summarizes the agent's own committed beliefs; it never reads ground truth. It restores the agent's ability to retrieve what it recorded; it does not correct what the agent recorded wrongly.

That distinction is measured, not asserted. On a model that computes its running state without drift, the compaction produces essentially full recovery, because the agent's committed snapshot happens to stay accurate. On a model that genuinely drifts, the informative signature appears: on the minority of runs where the agent commits a wrong value, the compaction carries that wrong value forward, and the managed error sits measurably above the truth-grounded floor. A mechanism that consulted an oracle would pin every run to the floor. It does not. The reduction therefore acts on the reflexive term itself, the reconstruction the agent no longer has to perform, and not by supplying the state the agent was meant to derive. It is the inequality's own prescription operating as the theory says it should, which is what separates a derived mechanism from a convenient one.

Where the result stands today, and the shortest path from a substrate result to a deployment claim

The evidence to date is drawn from a controlled substrate: an in-memory ledger in the executable-code action space that is the dominant coding-agent scaffold, instrumented so that the agent's committed state is directly inspectable and the reflexive term can be read without ambiguity. That substrate was chosen because it isolates B_ρ cleanly, and its retrieval-dominated character is also why the recovered fraction is so high and the residual so small. A production task carrying more genuine maintenance load, a larger H_ext, would leave more of the context requirement irreducible. The honest reading is scoped accordingly: the intervention reduces the reflexive share of the requirement, which on this substrate is most of it and on a harder task would be a smaller but still material fraction, and it does not reduce the share that is genuine task information.

The remaining step is transfer, and it has a defined shape rather than an open one. The head-to-head moves to public long-horizon benchmarks that are self-referential, where the context an agent must hold is its own accumulated history and the self-modeling burden is therefore present, rather than to retrieval benchmarks where the context is external reference material and no such burden exists. A benchmark of the first kind with a tunable context budget carries the managed and unmanaged comparison directly; a retrieval-only benchmark, run as a matched control, should show no comparable advantage from the same reduction, and that asymmetry isolates the reflexive term from ordinary difficulty at the level of the benchmark itself, the matched-twin logic raised one level up. Passing that test converts the context wall, the regime in which additional context stops helping and only reflexive-burden reduction moves the outcome, from a result on a substrate we operate into a result on agents and tasks we do not. A design-partner engagement runs in parallel and on the same principle, measured on the partner's own agent and task rather than on ours.

One component of that transfer is already secured. The committed-state re-surfacing on which this compaction rests, the spending of the budget on the state the next action presupposes rather than on the raw history, has now been shown to repair the coherence failure directly on real code, on a code-writing agent run on Amazon Bedrock and checked by executing its committed code, across three model families among them a frontier Claude and Amazon’s own; a two-step decoupling under the same record pressure does not recover it, which locates the effect specifically in the re-surfacing of committed state. That establishes the underlying mechanism and its repair on the deployment platform and against production models. It does not by itself establish the token-and-cost claim on real code, which still requires the self-referential-benchmark head-to-head set out above; what it removes is the prior question of whether the coherence phenomenon and its committed-state remedy survive off the controlled ledger, which they do. A first measurement of the cost side on the deployed real-code substrate, taken in real platform billing tokens rather than a proxy, now shows the expected qualitative relationship, the managed agent's context staying bounded as the session grows while the unmanaged agent's context grows every turn; this is reported as a measured scaling rather than a fixed multiple, and the fuller quantitative token-and-cost claim on real code still awaits the benchmark head-to-head.

That deployed instrument is, moreover, platform-agnostic. The same shared code core now runs, confirmed live, on Microsoft Azure AI Foundry as well as on Amazon Bedrock AgentCore, instantiated on each platform’s own native primitives: on Foundry the committed-state memory sits on a thread store, the deterministic verifier runs as an Entra-authenticated Azure Functions tool returning the identical pass-or-fail verdict as its Bedrock counterpart, and the coherence port runs as a hosted-agent container serving the same OpenAI-compatible response contract. For context efficiency this is more than a portability note. Because one and the same committed-state instrument can be placed onto two different platforms’ memory and harness layers, it opens a capability the field has so far had no clean way to exercise: measuring how much of the context-and-coherence cost of a given task is set by a platform’s own harness and memory design rather than by the model or the task itself. A platform that manages an agent’s history differently should, in principle, charge a different reflexive-burden bill for identical work, and a single instrument that ports across platforms on matched terms is what would make that comparison legible. The cross-platform coherence-cost comparison is the next measurement, to be run with matched models and tasks.

What this enables for a team operating agents at long horizons

For an organization running expensive long-horizon agents, one paying for very large context precisely because coherence degrades without it, the mechanism translates into a concrete engagement. It becomes possible to separate the portion of an agent's context that carries the reflexive burden from the portion that is genuine task information, to reduce the burden that is forcing the window open, and to demonstrate the agent holding its coherence at a shorter, and therefore cheaper and faster, context. The diagnostic capability that measures where and why an agent loses coherence is the proven near-term asset. Context efficiency is the compounding capability that sits on top of it, and the two results here are the first controlled evidence that the upside is real rather than rhetorical. As a rough sense of the magnitude, on the frontier model measured here the same-coherence budget is roughly a quarter of the raw-window budget in exact billing tokens, which at list pricing is on the order of a fourfold reduction in per-run inference cost; for a fleet of a million long-horizon runs a month that is on the order of two million dollars a year. That figure is illustrative at list pricing and scales with a deployment's own run cost and volume rather than being a fixed claim, but it locates the order of magnitude of the second lever alongside the reliability gain.

The measurement holds on the clouds enterprises deploy on

The evidence to this point is controlled and in vitro, run on the program's own substrate. A fair question from anyone fielding agents is whether the reflexive burden the theory names survives the move onto a real commercial platform, or whether it is an artifact of the measurement. To answer it, the same record-capacity instrument was harnessed onto the two clouds enterprises field agents on today, Amazon Bedrock and Microsoft Azure AI Foundry, and the record-pressure sweep was run through each. Two open-weight models that both clouds serve, a compact model and a larger reasoning model, were held fixed across the clouds, and each of the four model-and-cloud cells was measured as a full sweep at six independent runs per level.

The onset reproduced on both clouds, in the saturating shape the theory predicts: with task difficulty and interaction density held fixed, the burden rose from near zero to a plateau as the record window was starved. More telling than the reproduction was its uniformity. The four curves lay on top of one another, the same reflexive burden for both models and on both clouds, the same onset near a third of the horizon, and the same ceiling. The coherence boundary is a property of the work, not of the vendor underneath it, so an operator cannot buy past it by switching clouds or paying for a larger model. Reducing it, the second lever of this brief, remains the only move that changes it.

Where the clouds did separate was in ordinary task difficulty, the H_ext floor rather than the reflexive term, and there they separated by model: the compact model executed its grounded arithmetic roughly ten times more cleanly on one cloud than on the other, while the larger model was steady across both. That is the expected shape. The exogenous term belongs to the serving stack and varies with it, while the reflexive term is intrinsic and does not, and the instrument reads them apart, returning the difference as a number per model and per cloud rather than a vendor claim.

The broader implication for anyone building at long horizons follows from the same inequality that opened this brief. Context length has a ceiling in exactly the regime the frontier is selling into, and on the far side of that ceiling the only remaining lever is the reduction of reflexive burden. That the lever exists, that its form is dictated by the theory rather than guessed, and that it recovers most of the burden it targets across four model families, one of them a frontier model, is the result now in hand.