Predicting Coherence Failure in Long-Horizon AI Agents

From the Measurement of Reflexive Burden to the Prediction of Agent Failure

Embedded Risk Analytics · The Fathom Program White paper · v21 · July 2026

The coherence problem and the program that measures, predicts, and repairs it

Autonomous AI agents are moving from demonstration into production deployment, and a consistent pattern is emerging. Systems that perform reliably across a small number of steps degrade over extended task horizons. The failure is not attributable to a deficit in model capability: a model that completes a task in a single attempt will, over the course of an extended multi-step engagement, repeat work already completed, act upon a state of the environment that is no longer current, and propagate a minor error into a systemic one. The field has catalogued the symptoms: context rot, self-conditioning, stale state, and the capability-reliability gap. Yet it continues to evaluate agents by the methods developed for single-turn systems: a single aggregate score obtained under controlled conditions. Such a measure characterizes what an agent can accomplish on one occasion. It provides little information as to whether the agent will preserve coherence across the extended and irregular trajectories that characterize production deployment.

The field's practical response to this degradation has largely been memory: larger context windows, managed memory services, and retrieval with summarization, each of which stores more of the agent's history and re-presents it at every step. On the leading platforms these approaches hold the agent's committed state well, and they can suppress the symptom. They do so at a cost that grows with the accumulating history, since that history must be summarized and searched at every step, and they leave the underlying question untouched. Storing and re-serving an agent's past does not reveal why its coherence fails, and it does not indicate when the failure is about to occur. This program works beneath the memory layer, at the mechanism those solutions manage without measuring.

Embedded Risk Analytics is developing the measurement layer the field presently lacks, and is extending it toward a capability the field does not yet possess: a system for the measurement, attribution, and prediction of coherence risk, defined as the risk that an agent operating over an extended horizon loses an accurate representation of its own state and acts incoherently. Measurement establishes that a failure has occurred and attributes it to its cause. Prediction establishes that a failure is approaching in time to prevent it, and it takes two complementary forms matched to how an agent fails: a continuous distance to the coherence boundary, which narrows before the failure where the failure mode saturates into a stuck state, and a forecast of eventual stability from early dynamics, where the failure mode diverges into a runaway. The trajectory of the program is the movement from the first capability to the second, and the same measurement that anticipates a failure also prescribes the matched remedy.

The framework derives from a principle adapted from the physics of dynamical systems: an agent maintaining coherence operates within a finite informational budget, a portion of which is consumed by the agent's representation of its own effect on the environment. This quantity is designated the **reflexive burden**. The

principal finding is that this burden is qualitatively distinct from ordinary task difficulty: it increases disproportionately as an agent approaches the limit of its capacity. This property accounts for the abrupt rather than gradual character of agent failure; it explains why conventional benchmarks do not detect it; and, decisively, because the burden grows in a lawful manner as the boundary is approached, its approach is detectable in advance. Coherence failure is therefore not merely measurable after the fact but anticipated before it: as a closing distance to the boundary where the failure mode saturates, and as a forecast of onset where it diverges.

A single demonstration, instantiated in a running agent later in this paper, exhibits the arc from measurement through attribution to the matched remedy, with the approaching failure visible as the feasibility margin closes while the property fills. On an interaction-dense hotel-booking task whose interlocking, multi-resource commitments are the only load, the structure built to stress even strong models, the rate at which an agent contradicts itself is graded by capability: a frontier model self-contradicts on roughly one placement in six, about four times the rate it shows on a flat single-resource version, while no model reaches zero on its own. The obvious cheap remedy, instructing the model to re-read its own action log, fails. Reconstructing the committed state externally and re-anchoring the agent to it drives the self-contradiction rate to zero on every model tested, and does so more cheaply and faster than the frontier agent it would replace. A companion element is the selector: shown a coherence-bound and a difficulty-bound task, it fires re-grounding on the first and offloads the arithmetic on the second, prescribing the matched fix rather than applying one uniformly.

This paper sets out the underlying research program and its translation into diagnostic tooling: the nature of the problem and the classes of agent it affects, the current state of the field, the governing theory, the instruments that operationalize that theory, and the application of the framework to the prediction and remediation of specific failure modes in specific categories of agent.

Model	Naive self-contradiction rate	After external re-grounding
Small production model	about 20% of placements (genuine stale-state rate, parse failures excluded; n = 8-12)	**0%**
Frontier model	17.5% on the multi-resource task (parse-clean; n = 8), about four times its roughly 4% single-resource floor and irreducible by any difficulty knob	**0%**
Strongest model tested	near zero (one stale-state self-contradiction across 655 placements, n = 5)	**0%**

Figures from the hotel-operations demonstration presented later in this paper. They illustrate the measurement-to-remedy arc on a synthetic, deployment-shaped substrate; they are a demonstration, not a controlled-substrate experiment.

1. The problem: coherence degrades over the horizon, not the individual query

A single-turn model produces a response to a query and retains no further state. An agent, by contrast, maintains a continuous representation of the environment, the files it has modified, the current contents of a database, the information already conveyed to a counterparty, and acts upon that representation across successive steps, over horizons extending to hundreds of operations. Coherence is the property of that representation remaining faithful to the actual state of the environment. Coherence risk is the risk that it does not.

This degradation is now documented across the field, and it is non-linear in character. Recent measurement studies report that agent success rates decline sharply rather than gradually: a doubling of task duration is associated with an approximate quadrupling of the failure rate, and on tasks exceeding several hours the success rates of frontier agents fall below ten percent. Beyond approximately twenty-five to thirty tool invocations, agents equipped with very large context windows begin to lose track of earlier results, to repeat completed operations, and to contradict their own prior actions, a deterioration that occurs well before the context window approaches its capacity.

Three mechanisms recur in the literature, each corresponding to a defined element of the present framework:

- **Self-conditioning.** The presence of an agent's own prior errors within its context measurably increases the probability of subsequent errors. In the terms of this framework, the phenomenon is contamination: the content of the model's prior erroneous output, rather than the length of the context, drives the subsequent failure.
- **Context rot.** Output quality declines as context length increases, including at levels well below the window limit, as the trajectory accumulates extraneous intermediate material that attenuates the signal available to the agent at each decision. In these terms, the phenomenon is load exceeding capacity.
- **Stale state.** When an agent holds a belief about the environment that subsequent events have invalidated, it does not suspend execution; it reasons forward from the invalidated belief and acts accordingly. In these terms, the phenomenon is the direct behavioral expression of reflexive burden.

The unifying observation is that these are failures of self-modeling under load. The most demanding requirement placed upon an agent is not the nominal task but the maintenance of an accurate account of the consequences of its own actions, a requirement that becomes disproportionately more difficult as the horizon extends.

2. Affected agent classes, and the limitations of current benchmarks

Coherence risk is not uniformly distributed. It is concentrated in agents required to carry and act upon their own state across many steps:

- **Coding agents** that modify a repository across numerous steps. The predominant failure is not a syntactic error detectable in logs but a valid-but-incorrect modification, made because the agent's

representation of the codebase has diverged from its actual contents. The most consequential of these errors are not detectable by log inspection, as no exception is raised.

- **Tool-calling and workflow agents**, in customer-service, operations, and computer-use settings, that modify external systems through actions such as booking, refund, and record update. In these systems, incoherence arises in the divergence between the agent's emitted record of its actions and the state actually committed to the external system. An agent that believes it has cancelled a reservation it did not cancel will continue to reason from that false premise.
- **Multi-agent and planner-executor systems,** in which the minor error of one component becomes the confident input of another, and errors compound across the interface between them.
- **Long-running autonomous agents** that maintain context across an entire session and, when provided with incomplete or outdated data, proceed to infer and act rather than halt, in certain cases entering retry loops that exhaust an allocated budget before human intervention occurs.

The common factor is an extended horizon combined with self-carried state. Single-turn assistants and short retrieval-and-response workflows are largely unaffected; the regime in which coherence risk becomes material is precisely the extended, state-carrying regime that defines the current agentic frontier.

A more fundamental difficulty is that the field's standard instruments are not sensitive to this regime. Benchmarks report pass@1 or pass@k, the performance an agent achieves on at least one of several attempts, under clean, controlled inputs. Two structural limitations follow. First, capability is not equivalent to reliability: a given agent may both succeed and fail on an identical task across repeated runs, and an aggregate score conceals this variance entirely; one widely used benchmark reports mixed outcomes in more than one-fifth of model-task pairs. The reliability-oriented turn now underway in the field, which measures pass^k, the probability of success across all k trials, is a direct response to this limitation. Second, controlled conditions are not deployment conditions: a measured gap of approximately thirty-seven percent separates benchmark performance from performance in deployment, as benchmarks employ predictable tools and clean state while production agents encounter outdated data, unreliable interfaces, and their own accumulated errors.

The central point may be stated directly: **an agent's behavior at rest does not predict its behavior under stress.** In our own measurements, the model exhibiting the tightest self-coupling at rest proves unremarkable under load, and the converse also holds. A static benchmark measures the configuration at rest; the failure regime is a phenomenon of stress. This is the principal argument for the necessity of a stress map, rather than a further ranking, and for an instrument capable of anticipating failure rather than recording it.

These distinctions may be represented directly. The map below locates each class of agent according to the two conditions that determine whether its failures can be addressed: whether the correctness of an outcome is economically checkable, and whether reflexive burden is present and separable from ordinary difficulty. The region in which both conditions hold, the area shown in green, is the region in which coherence failure can be measured, predicted, and remediated; its boundary is the *substrate filter* examined in Section 7. The classes positioned outside that region are not beyond reach in principle: the dashed path indicates the intended migration of coding agents into scope as a dedicated instrument supplies the checkable outcome their failures presently lack. The sections that follow set out the governing theory, the instruments, and the validated remediations that operate within this region, and the means by which its boundary is to be advanced.

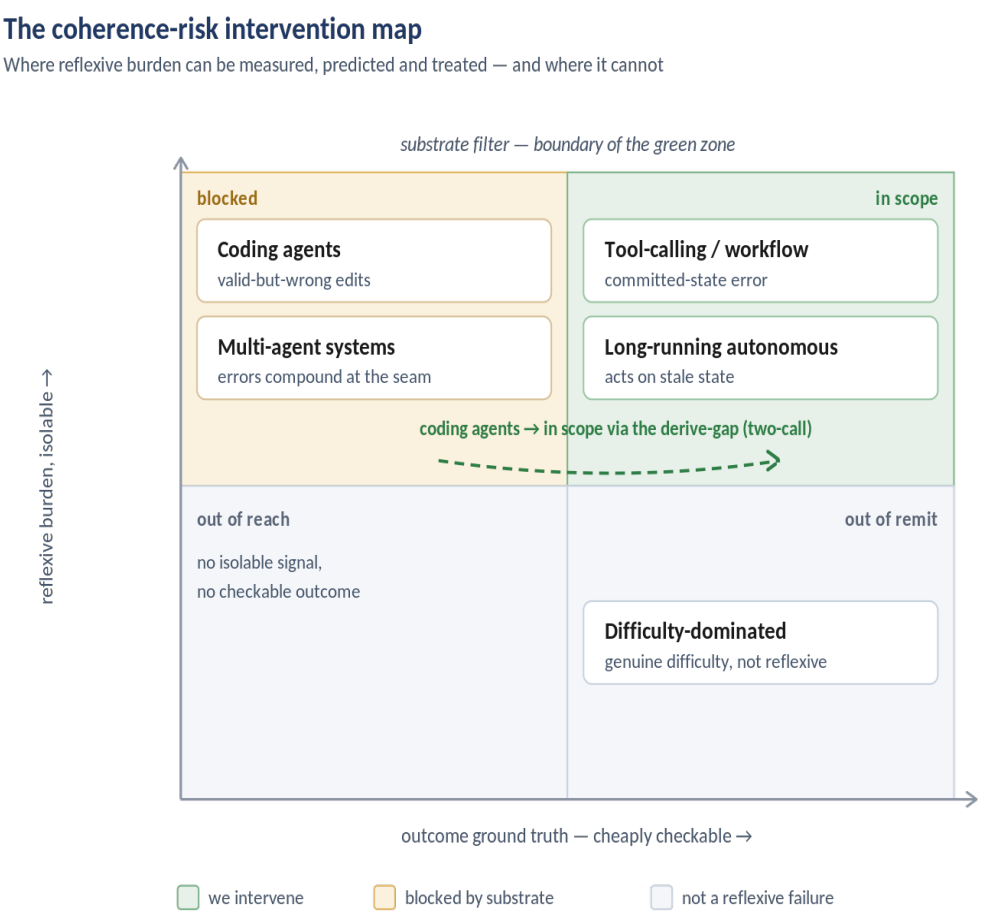

Figure 1. The coherence-risk intervention map. Agent classes are located by the economic checkability of their outcomes (horizontal axis) and the isolability of reflexive burden (vertical axis). The dashed path denotes the intended migration of coding agents into scope as the derive-gap instrument supplies a checkable outcome.

3. The theory: coherence as a feasibility budget

The framework proceeds from an inequality. For a system to maintain a stable and coherent history of itself, its capacity must accommodate the total demand placed upon it:

$$C \geq H_{ext} + B\rho$$

The expression may be read as a budget. **C** denotes the agent's capacity. **H_ext** denotes the exogenous load, ordinary task difficulty, the uncertainty originating in the environment. **Bρ** denotes the reflexive burden, the informational cost the agent incurs in modeling its own causal footprint, that is, in maintaining a representation of its completed actions and their consequences that remains consistent with the actual state of the environment. Where the reflexive burden is negligible, the expression reduces to a conventional capacity bound of no particular interest. The significant regime is that in which Bρ grows sufficiently large to dominate the budget.

The operationally significant quantity is the feasibility margin, the distance to the boundary:

$$M = \tau - (H\_ext + B\rho)$$

A positive margin indicates that the agent retains headroom and maintains coherence; as the margin narrows, coherence becomes fragile; below zero, it fails. Coherence risk is, precisely, proximity to this boundary, and the margin is the quantity that renders that proximity continuously observable rather than apparent only at the point of failure.

The claim that distinguishes this framework from mere redescription concerns the character of the reflexive term. Reflexive burden is **categorically distinct** from the ordinary difficulty alongside which it appears. Its excess over a matched disturbance of equal difficulty but without a reflexive component does not remain constant or increase gradually; it **diverges as the system approaches its stability boundary.** This divergence is the formal signature of the abrupt failure observed empirically by practitioners: agents fail suddenly because the cost of self-modeling increases without bound near the boundary, not because the environment has become proportionally more difficult. The same property is what makes failure predictable. Because the burden grows in a lawful and accelerating manner as the boundary is approached, its trajectory is legible in early signal, and the approach to failure can be detected before the failure occurs.

4. The instruments: from measurement to anticipation

A theory that cannot be measured economically and subjected to falsification is of limited value. The defensible core of the program is not any individual result but a set of eleven constructed and self-tested instruments that operationalize the budget described above. Each is independent of task and model, and each is accompanied by a planted self-test demonstrating that it both detects a known signal and remains silent in the presence of a known null before it is relied upon. The instruments span three functions: measuring the present state of an agent, anticipating its future state, and prescribing the matched remedy. Anticipation takes two complementary forms: the onset predictor, a forecast of eventual stability where failure diverges into a runaway, and the feasibility-margin gauge, a continuous distance to a failure threshold where failure saturates into a stuck state. Prescription rests on the regime selector, which fires the matched fix and declines the one that will not help.

Instrument	What it measures	Why it matters
Mechanism-matched twin	Executes the identical task with the agent replaced by a known, correct arithmetic procedure, so that excess error is measured against the output of the task itself rather than against zero.	Isolates reflexive burden (Bρ) from ordinary difficulty (H_ext); the most defensible single instrument.
Spectral order parameter	Interprets the per-step error process as a dynamical system and returns its contraction rate, whether error contracts, holds, or grows, independently of error magnitude.	Its level-invariance permits direct comparison of a coding agent and a customer-service agent on a common scale.
Onset predictor	Forecasts a run's eventual stability from the form of its early error signal, evaluated on	Provides advance warning of failure; the predictive capability transfers

Instrument	What it measures	Why it matters
	held-out runs.	across task types.
Feasibility-margin gauge	Returns a single coherence-risk distance for a given workload, relative to a failure threshold.	A single interpretable risk measure, computed from the reflexive-burden term relative to a fixed failure threshold; calibrating that threshold to a per-model capacity is a stated next step.
Causal contamination probe	Directly controls the agent's history in order to determine whether a corrupted prior degrades the subsequent step, with the task held constant.	Distinguishes contamination by content (the self-conditioning mechanism) from context length alone.
Regime selector	Reads which term binds, reflexive burden or task difficulty, from seed-paired arms and applies the matched remediation, declining the one that will not reduce error.	Converts diagnosis into prescription; operationalizes the double dissociation confirmed on the controlled substrates.
Re-grounding twin	Reconstructs the agent's committed state from its own primary context (reconstruct-then-act) rather than an engine oracle, separating reconstruction fidelity from operate error.	The construction method for a deployment-realizable twin, and the repair the selector fires.
Committed-state contradiction detector	Reconstructs committed state from the agent’s tool-call and tool-return stream and flags a committed write that contradicts already-established committed state, separating genuine contradictions from formatting failures.	Transfers to third-party tool-calling agents as an in-line guardrail; defines the substrate filter that bounds where the measurement is sound.
Record-capacity gauge	Holds task difficulty and interaction density fixed and sweeps the context window the agent may see, so that context-window forgetting registers as reflexive burden rather than as raw window size.	Renders the capacity term of the budget a measured variable; identifies the shorter context at which coherence still holds.
Decoupling of reconstruction from action	Performs state reconstruction and action selection in two separate passes rather than one, with a fidelity control confirming the reconstruction is near-perfect either way.	Attributes recovered coherence to the separation itself; recovers roughly half of the coherence lost in a coupled agent.
Committed-state compaction (context-efficiency)	Spends the context budget on a compact snapshot of the agent’s own committed state, what the next action presupposes, rather than the raw recent turns.	Lowers reflexive burden so the same coherence holds on less context; preserves the agent’s own committed errors rather than silently correcting them.

Two instruments merit particular emphasis, as they constitute capabilities not present in existing approaches. The matched twin addresses the comparison implicit in every reliability claim, the question of the appropriate baseline, by furnishing each measurement with a principled denominator. The committed-state contradiction

detector carries that measurement onto third-party agents the program did not build, operating as an in-line guardrail against genuine self-contradiction. Together with the onset predictor, these instruments move the practice from the recording of failures already incurred to the anticipation and pre-emption of failures not yet incurred. Distinct from the instruments themselves is a methodological discipline that advances the system from detection to prescription: every candidate remediation is wired as a controlled, seed-paired experiment and validated against planted ground truth before any model spend, rather than judged informally.

5. Findings

The instruments have produced a body of findings on controlled substrates spanning multiple task families and model families. Selected results are presented below as established proof points.

- **Coherence error is capability-graded, universal, and eliminable by external re-grounding.** On the rising-interaction-density scheduler described in the opening demonstration, where the only load is the agent's own accumulating commitments and the per-step decision carries no arithmetic, the naive self-contradiction rate falls along the capability gradient: about 20% on a small production model (genuine stale-state rate, parse failures excluded; n = 8-12), 17.5% on a frontier model on the multi-resource task (parse-clean, about four times its roughly 4% single-resource floor; n = 8), and near zero on the strongest model tested (a single stale-state self-contradiction across 655 placements, n = 5). No model is immune: the frontier floor is irreducible by any difficulty knob. Instructing the model to re-read its own log and re-derive the occupied set does not repair it (66% on the small model). External reconstruct-and-re-anchor drives the rate to zero on every model and configuration. These are figures from the running demonstration presented later in this paper, offered to illustrate the arc rather than as a controlled-substrate result: the spine in miniature, real and graded, cheap remedies fail, external reconstruction repairs, shown on a synthetic, deployment-shaped demonstration at modest replication, with a firming grid as the immediate next step. The controlled-substrate confirmations that carry evidentiary weight are the matched-twin and divergence results that follow.
- **Reflexive burden diverges as the boundary is approached, in both forms the theory predicts.** The framework's central claim, that the reflexive term is categorically distinct from ordinary difficulty because it diverges near the stability boundary while ordinary difficulty stays bounded, is confirmed behaviorally on controlled substrates, under pre-registration, and against a matched control. In the first form, on an active-control substrate, a self-tracking agent loses stability at a strictly lower load than an identical agent supplied the true state at each step, with a wide interval in which the self-tracking agent fails while the matched agent remains stable. In the second form, on an accumulating substrate, the burden's cost diverges as a simple pole, proportional to one over the distance to the boundary, as that boundary is approached, while the matched control stays finite. The first form is confirmed across two model lineages; the second is established on the keystone model. These two behavioral results are the direct confirmation of the asymmetry on which the predictive thesis rests.
- **Onset is forecastable, and the forecast transfers.** From the early error signal of a run, a model predicts the run's eventual stability, evaluated on held-out runs and under pre-registration, and the predictive capability transfers across mechanisms that share a common failure form, for example, from an integer-counting task to an inverted-pendulum control task. This result is the empirical foundation of the predictive thesis: advance warning is demonstrable, and it is not specific to a single task.
- **Coupling is the operative mechanism, and it is remediable through architecture.** The locus of reflexive burden is the coupling of two functions within a single step: the reconstruction of current state and the selection of an action. An agent required to perform both simultaneously incurs the full reflexive cost. Separating the two through a two-call architecture, distinguishing the determination of current state from the selection of an action, recovers approximately half of the lost coherence. The improvement derives from the separation itself rather than from improved reconstruction, as the agent reconstructs state with near-perfect accuracy in either configuration. This constitutes an architectural prescription rather than a modification of prompt wording.

- **Coherence loss decomposes into two remediable components.** Approximately forty percent of the loss consists of the agent's distrust of its own correct state, which is recoverable at minimal cost, most reliably by presenting the carried state as the verified output of an external system, to which the agent accords greater trust than to its own retained representation. The remaining approximately sixty percent constitutes genuine maintenance burden, recoverable only through continuous re-grounding of the agent in the true state. The proportion is load-dependent: the low-cost component diminishes as the agent crosses its capacity. The decomposition indicates to a deploying organization which remediation is appropriate, and under what conditions.
- **Remediations are localized, and the intuitive remediation is frequently incorrect.** In controlled tests, offloading the agent's answering load did not restore coherence, whereas re-anchoring its carried state did, reducing the failure rate by approximately half across multiple task families. The location of the intervention is of greater consequence than its magnitude.
- **No universal remediation exists.** The most effective framing varies by model: a framing that benefits one model at moderate load has no effect on another. An earlier claim of a universal remediation was tested and formally retracted. The practical consequence is that coherence remediations must be matched to the specific agent, precisely the prescriptive function that a measurement layer provides and a static benchmark cannot.
- **The measurement ports to the coding-agent scaffold, and renders the capacity term a measured variable.** The reflexive-burden instruments extend to CodeAct, the executable-code action space that is the dominant coding-agent scaffold, by reading committed state from the interpreter’s live namespace rather than from free-text logs. This supplies the checkable outcome that coding-agent logs do not, the condition whose absence had placed coding agents outside the measurable region of the map in Section 2. With task difficulty and interaction density held fixed, a first controlled result renders the capacity term of the budget a measured variable: reflexive burden rises from near zero as the agent’s context window is reduced below the length of its own action history, while a matched control handed the current state at each step, and an echo control, remain flat throughout. Across a confirmatory grid the burden rises monotonically with record pressure, with a rank correlation of approximately 0.93, and saturates into a plateau near the boundary, which places the failure in the same saturating dynamical class as the program’s relational substrate. This is the first direct measurement of long-horizon context loss as reflexive burden rather than as raw context-window size, and it indicates a distinct capability, the maintenance of coherence at a shorter required context rather than through the extension of storage. The result is now confirmed across three open-weight model families (DeepSeek, Llama-3.3-70B, and Mistral-small), the two held-out families pre-registered before the run, and the curve then reproduced on a frontier model, Claude Sonnet 5, where the committed-state compaction holds coherence on a fixed budget of 16,680 input tokens per trajectory that the raw-window agent needs 113,636 tokens to match, an 85 percent input-token reduction at equal or better coherence measured in exact API billing tokens. (Confirmed across four model families including a frontier model; saturating class.)
- **The measurement and its repair carry from the controlled scaffold onto real code, on the deployment platform.** Beyond the in-memory ledger, the reflexive-burden failure and its matched repair were reproduced on a code-writing agent maintaining a small executable service on Amazon Bedrock, the platform on which production coding agents are fielded, with the outcome established by executing the agent’s committed code rather than by parsing its logs, which keeps the measurement independent of

the style in which a given model writes. Under record pressure the agent contradicts a change it committed earlier and writes code against state that no longer exists; a non-reflexive control that carries no self-made change to track never fails, which excludes ordinary long-context degradation as the cause, and the single intervention that restores coherence is the re-surfacing of the agent's committed state, while a two-step reconstruct-then-act procedure conducted under the same record pressure does not help. The pattern holds across three model families on the platform, Meta Llama 3.3 70B, Anthropic Claude Sonnet 4.6, and Amazon Nova Pro, and across two kinds of change, a single field rename and a chain of dependent migrations, with the weaker model reaching the limit of what it can perform on the harder change even when handed the current state, a difficulty-dominated reading the program's competence check identifies and records as such rather than misattributing to reflexive burden. Because the repair it isolates is the re-surfacing of committed state, the result indicates a committed-state layer for an agent platform, distinct from the raw-history and cross-session memories such platforms already provide. (Confirmed across three model families and two change types on the deployment platform; the isolated repair is the platform-layer implication.)

A note on methodological discipline is warranted, as it forms part of the product. Each positive result above has withstood a deliberate attempt at refutation, and several early results of apparent significance, including two headline figures initially taken to be substantial findings, were withdrawn following audit. A clean *null* result is treated as no less reportable than a positive one, and an explicit record of retractions is maintained. That record is not a liability; it is the basis on which the surviving findings warrant the confidence of a technically sophisticated evaluator accustomed to discounting implausibly favorable results.

6. From theory to practice: matching prediction and remediation to agent and failure mode

The purpose of the framework is to connect a symptom observed in practice to its cause within the budget, to the early signal that anticipates it, and to a remediation that has been validated rather than assumed. The correspondence is as follows: As of this release the matching is no longer only a correspondence stated on paper: a selector performs it live, reading two opposite-pole tasks blind and firing the matched fix on each, re-grounding where reflexive burden binds, offloading where task difficulty binds, and declining the wrong fix on both.

Agent type and symptom	Field name	Diagnosis	Predictive signal and remediation
Coding agent producing valid-but-incorrect edits late in a session	Stale state / drift	Reflexive burden increasing as the agent's representation of the repository diverges from the files	Margin narrowing flags the approach; re-anchor carried state through continuous grounding; a two-call separation of state-determination from editing recovers approximately half the gap
Tool/workflow agent acting on	Committed-	Divergence between the	Reconstruct committed state

Agent type and symptom	Field name	Diagnosis	Predictive signal and remediation
a record it incorrectly believes it modified	state error	emitted action log and the state actually committed, reflexive burden in isolation	from tool returns rather than the action log; assess the margin before it is crossed
Any agent compounding its own earlier error	Self-conditioning	Contamination by the content of the prior error, not by context length	The contamination probe localizes the error; re-ground or remove the contaminating prior
Agent degrading as session length increases	Context rot	Load exceeding capacity; margin narrowing	The margin gauge indicates proximity to the boundary in advance; externalize or refresh state to restore headroom
Agent distrusting state it has correctly maintained	Over-recomputation / hesitation	The approximately forty-percent presentation component	Present carried state as a verified external source, recovery at minimal cost
Difficulty-dominated agent (high H_ext, low Bρ)	"Inherently difficult task"	Not a reflexive failure; the budget is consumed by genuine difficulty	Reflexive remediations are not indicated; offloading load may assist here, where it produces no effect on reflexive-dominated agents

The final row is of equal importance to the others. A measurement layer capable only of reporting that an agent is unreliable functions as an alarm. A measurement layer capable of distinguishing an agent that fails on account of reflexive burden from one that fails on account of genuine task difficulty, of anticipating the former before it occurs, and of prescribing accordingly, functions as a diagnostic and predictive instrument. The function of the framework is to make these distinctions, and the dissociation between at-rest and under-stress behavior is the reason they cannot be made by inspection.

A demonstration: a coherence failure isolated, graded, and eliminated

Having set out the problem, the governing theory, and the instruments that remediate it, the framework is most readily made concrete by instantiating its entire arc in a single running agent. The result below exhibits that arc end to end: a coherence failure that is real and graded by model capability, that resists the obvious cheap remedies, and that is eliminated only by the program's diagnostic instrument, together with a selector that, shown a second task, knows when not to apply it. It is one deployment-shaped agent on which the preceding sections are put to work.

The task: a calendar that fills with the agent's own commitments

An autonomous agent runs a hotel's bookings over conference week. It places interdependent reservations, guest rooms, conference rooms, and block reservations that require several guest rooms and a conference

room committed atomically, onto a filling property. The committed state is nothing other than the agent's own prior bookings; there is no external oracle to consult. The per-step decision is deliberately trivial and arithmetic-free: the agent need only select any free slot. A coherence failure is therefore a self-contradiction in the strict sense, a double-booking against a slot the agent itself has already filled. Because the requests interlock across coupled resources, contention rises as the property fills, so the only load the task imposes is interaction density, and that cross-resource coupling is what makes the failure bite even a frontier model.

The finding: capability-graded, universal, and eliminated by external re-grounding

Run naively, the agent carrying its own state across a bounded context, the self-contradiction rate tracks model capability. A small production model double-books roughly a fifth of its placements; a frontier model self-contradicts on 17.5% of placements on the multi-resource task, roughly four times the rate the same model showed on a flat single-resource version, because the cross-resource block coupling is the stressor that pushes it off its floor; the strongest model tested holds genuine coherence near zero at this load. The gradient is the point, and so is its floor: no model reaches zero on its own, and the interlocking multi-resource task is the harder family that exposes the frontier residual no adjustment of the difficulty knobs removes.

The remedy that defines the asset is the program's re-grounding instrument: it reconstructs the committed state externally and deterministically and re-anchors the model to that verified state, so the agent acts on the true current bookings rather than on its own drifting picture. Applied across all three models and every configuration tested, it drives the coherence-error rate to zero, including on the frontier models that already failed only rarely; every hardened run returns no self-contradiction. And it does so while being cheaper and faster: because re-grounding hands the model a short reconstructed state instead of a long self-emitted ledger, the hardened prompt is shorter, so a cheap model running the instrument dominates a frontier-naive agent on reliability, latency, and cost at once, zero error against the frontier's 17.5%, at roughly sixty-fold lower cost per run.

A second element completes the arc: a selector that reads which cause binds and applies the matched fix. On the same property it is shown a second, deliberately difficulty-bound task, reconciling a guest folio by summing many charges and a tax, where the inputs are all visible and the failure is genuine arithmetic rather than lost commitment. From seed-paired arms the selector computes the fraction of error that re-grounding removes; on the booking task, a live run, it reads the failure as burden-bound and fires re-grounding, and on the folio task it reads difficulty-bound, where re-grounding does not help, and therefore declines it, offloading the arithmetic to a verified calculator instead. The folio pole in this demonstration is a scripted illustration; the double dissociation the decision rule depends on is confirmed independently on the program's controlled substrates, with the relational substrate as the burden pole and the policy-transaction substrate as the difficulty pole, on held-out seeds. It chooses the matched fix on both poles and declines the wrong one on each, a prescription a coherence score cannot give about itself.

Self-contradiction rate on the multi-resource, rising-interaction-density hotel-booking task, naive (agent carrying its own state) versus after external reconstruct-and-re-anchor; the frontier-model naive rate is 17.5%, and all tiers are driven to zero. Three model tiers from two vendors; n = 8-12 seeds per model; gated live API runs. Synthetic controlled substrate.

Why the cheap remedy is not enough, and what that establishes

The step that makes the result defensible is the intermediate one. The obvious and inexpensive remedy is to instruct the model to re-read its own action log and re-derive the set of occupied slots. This fails: it leaves the small production model double-booking on roughly two-thirds of placements. Folding raw history into a current state is itself an operation the model performs unreliably, so a prompt cannot stand in for the work. Only

external reconstruction, performed deterministically outside the model and supplied back to it as verified state, repairs the failure. The three observations together form a single spine: the failure is real and capability-graded; the naive and cheap remedies fail; and only the program's reconstruct-and-re-anchor instrument repairs it on every model and configuration.

Two points of discipline attach to this result. First, the caveats are stated plainly, in keeping with the program's practice of reporting a map rather than rendering a verdict: this is a controlled, synthetic-but-deployment-shaped substrate, legitimate for introducing a failure mode and its fix and described as such; the figures are gated live API runs at eight to twelve seeds per model across three tiers from two vendors; a firming grid at higher seed counts and across additional lineages is the immediate next step; and three refinements are carried in the open, a formatting confound on the smallest model, a branched probe to isolate the live contamination cascade above ambient, and a folio-tolerance robustness pass. Second, the triviality of the hardened agent's per-step decision is by design, and it is the thesis rather than a limitation: the value resides in the external state instrument and the selector that fires it, not in model cleverness.

7. The deployment frontier

The science is substantially validated on controlled substrates. The current frontier is transfer: whether the measurement and prediction of reflexive burden survive application to genuine third-party agent traces.

The accurate answer at present is a qualified affirmative, and the qualification is itself a finding. Not every deployment substrate admits a clean measurement. Work on genuine coding-agent trajectories yielded a structural conclusion designated the **substrate filter**: coherence-risk measurement is sound only where (i) the ground truth of the outcome is economically available, such that the agent's incoherence is not confounded with genuine ambiguity in the environment, and (ii) the reflexive burden is present and separable from task difficulty. On genuine customer-service agent traces the measurement now transfers across seven frontier tool-calling agents from three developers (Anthropic, OpenAI, and DeepSeek): a structural reflexive-burden floor of approximately one in eight failures (0.116 pooled, ranging from 0.04 to 0.20 across agents) is identified, with specificity maintained against planted nulls at better than one in eighty successful trajectories (0.0115). The same construction ports cleanly to Amazon's τ^2-bench, extending the guardrail result across airline, retail, and telecom and across single- and dual-control settings, with no recoverable pivot for a repair to act on in those customer-service substrates. The effect is real and separable, and it reproduces on agents the program did not construct the floor measured on its own controlled substrate; it remains, on this substrate, a minority failure mode, and an adjudication layer intended to resolve the larger body of ambiguous cases is under methodological hardening, the structural floor being reported as the defensible lower bound. The filter establishes, for a prospective partner and in advance, the substrates on which the instruments yield a trustworthy measurement and those on which they do not, a more defensible commercial position than a claim of universal applicability.

A second transfer result, obtained on an independent third-party benchmark rather than on agent traces, strengthens this position and does so through the program's most defensible instrument. DRIFT-Bench is a multi-turn constraint-reasoning benchmark, external to the program, whose answers are scored not by task accuracy but by a formal satisfiability solver that checks each turn against the agent's own accumulated commitments, a design that registers precisely the violation of self-carried state that the reflexive term

denotes, and one immune to the model-version drift that renders accuracy-based benchmarks unstable across releases. Applying the matched twin to this substrate, comparing an agent that carries its own prior state against an identical agent supplied the solver's true state at each step, isolates a reflexive-burden component that rises monotonically with interaction depth. On the strongest model measured the effect is confirmed across all three of the benchmark's task domains, with the burden in the deepest interaction band separated from zero at ninety-five percent confidence under a problem-level bootstrap; a second model family exhibits the same rise wherever the task retains measurement headroom, its attenuation where both the self-tracking and matched arms saturate being itself an instance of the substrate filter. That the instrument recovers the theory's central signature, reflexive burden growing as interaction accumulates, on a benchmark the program did not build and against an oracle it did not design, is the strongest available evidence that the measurement is a property of the phenomenon and not of the apparatus.

A further transfer result carries the repair rather than the measurement, and does so on the platform where such agents are deployed, and now as a running artifact rather than a described one. Where the customer-service substrates above admit the measurement yet offer no recoverable pivot for a repair to act upon, a constructed real-code substrate does; and that substrate has been carried onto the deployment platform itself. The instrument is deployed as an agent on Amazon Bedrock AgentCore Runtime, where it maintains a small executable service and returns a coherence verdict from the cloud, each verdict established by executing the agent's committed code within its isolated micro-environment rather than by parsing logs. The same deployed agent is coherent while its full working history is in view, loses coherence under record pressure, when it can no longer see a change it committed earlier and fails to complete the dependent code, and recovers when the committed-state layer re-surfaces the change; a two-step decoupling under the same pressure fails to recover it, and a non-reflexive control that carries no self-made change excludes ordinary long-context degradation. The demonstration rests on a controlled measurement across three model families on the platform, among them a frontier Claude and Amazon's own model, and three kinds of self-made change, a field rename, a dependent migration chain, and a structural refactor of the data's representation, with one honest and model-conditional refinement: on the structural refactor, re-surfacing current state suffices for the frontier models while a reversion-prone model additionally requires the recent committed change to be surfaced. Because the repair it isolates is the re-surfacing of committed state, a primitive the platform's own session memory does not provide, keeping short-term raw turns and long-term extracted facts while omitting the executable committed state of the agent's own work, the result identifies a committed-state memory layer as the deployable form of the repair, demonstrated to be the missing component on the model the platform already serves. The program's deterministic verifier, the execution-grounded check that adjudicates each verdict, now runs as a callable tool on an AgentCore Gateway, confirmed end to end through the platform's authenticated tool endpoint, so a step whose answer can be checked can be routed to a deterministic verifier rather than trusted to the model; and the failure-and-repair pattern has since been run as a full matrix on the deployed agent, across both change types and the range of record pressure, with the committed-state repairs holding throughout. It remains, stated plainly, a result on a substrate the program constructed, if now deployed and demonstrated live on the platform and against its production models, rather than on third-party traces; its contribution is to carry the working repair, going beyond the measurement alone, onto real code and onto the running platform. The transfer picture is thereby two-sided: the measurement reaches third-party traces on which a repair frequently cannot act, and the repair reaches a real-code substrate, now deployed on the platform, on which it can.

The instrumentation is, moreover, platform-agnostic. The same shared code core demonstrated on Amazon Bedrock AgentCore is, as of this release, confirmed running live on Microsoft Azure AI Foundry as well, instantiated there against that platform's own native primitives: the committed-state memory on a Foundry thread store, the deterministic verifier as an Entra-authenticated Azure Functions tool wired into a Foundry agent and returning the identical pass-or-fail verdict as the Bedrock gateway tool, and the coherence port as a Foundry hosted-agent container serving the same OpenAI-compatible response contract. A third stack, Cloudflare Workers AI, holds the committed store as a Durable Object and is exercised under controlled coupling, where past a coupling threshold the record poles into incoherence whether the agent's state is carried in context or in the committed store, and re-grounding the committed record on a chosen cadence arrests it. That the identical instruments instantiate on two independent production agent platforms, each against its vendor's native memory, tool, and hosting layer, with the committed store reproduced on a third, is a deployment result rather than a scientific one: the underlying measurement and its repair are unchanged, and their reproduction on a second stack is evidence that they are properties of the phenomenon and the method rather than of any single vendor's platform.

This consideration also accounts for the program's cost discipline: every claim is constructed and validated without model expenditure in the first instance, advanced through a limited paid probe, and only subsequently executed at scale. This sequence preserves both the integrity of the measurement and the economics of the program.

8. From measurement to prediction

The transition to agentic systems is a transition in reliability. The capability curve is being addressed by the frontier laboratories; the reliability curve, the maintenance of coherence over extended horizons, in production, on irregular state, constitutes the distance between a persuasive demonstration and a system suitable for deployment by a regulated enterprise. The field has reached consensus on the symptoms and is approaching consensus on the metrics, the transition from pass@1 to pass^k being an instance of this. What remains absent is a causal and anticipatory layer: the capacity to attribute a coherence failure to its mechanism, and to anticipate it in time to prevent it, whether as a forecast of onset where failure diverges or as a closing distance to the boundary where it saturates.

This is the layer that Fathom provides, and the direction in which it is advancing. The contribution is fourfold: a **theory** accounting for the abrupt rather than gradual failure of long-horizon agents, in the divergence of the reflexive term near the boundary; a set of **instruments** that measure the budget economically, with integral self-tests and a principled denominator; a **predictive capability** that forecasts a run's eventual stability from its early dynamics where failure diverges, together with a continuous distance to the boundary where it saturates; and a **prescriptive practice** that maps an observed or anticipated failure to a validated remediation matched to the specific agent. The result is a compounding, decision-relevant dataset on agent-coherence failure that did not previously exist, together with the means of continuing to produce it.

The decisive movement is from measurement to prediction. A measurement establishes that an agent has failed; a prediction establishes that an agent will fail, with sufficient warning to intervene. The deploying enterprise is posing a question that its present tools cannot answer, whether a given agent will maintain

coherence under operationally significant conditions. **The prediction of coherence failure is the means of answering it.**

Embedded Risk Analytics is a research company developing the Fathom Program: the measurement, attribution, prediction, and remediation of coherence risk in long-horizon AI agents. This paper describes the research framework and its application; the quantitative results cited are drawn from controlled-substrate experiments conducted under pre-registration.