

LOBSTER-Bench

A Long-Lived Agent Observability Benchmark
for Persistent AI Societies

Chen Ho Yiing

Independent Researcher, Taipei

ORCID: 0009-0006-6816-9891

`norika@charenix.com`

May 2026

Abstract

Most agent benchmarks ask whether an agent can finish a task. They reward tool use, web navigation, planning, retrieval, and workflow completion. I argue that this question is too narrow for the systems being built right now. The agents arriving in 2026 are not session-scoped task performers. They persist for days and weeks, accumulate memory, influence each other, drift, overload, and require governance. A benchmark that ignores time misses where these systems actually fail.

I propose LOBSTER-Bench, a benchmark that scores an agent system on six dimensions: temporal persistence, cognitive telemetry coverage, relational observability, collective task assay, cognitive load management, and governance and auditability. I anchor each dimension in measurements from the Lobster Observatory substrate, a multi-agent system I have operated for the past two weeks, comprising one archival agent plus twenty active agents, exposing fourteen core telemetry channels per agent, generating 27,788 dread-and-violation telemetry rows over a seven-day window, recording 6,354 settled wagers across five domains, and producing 2,546 documented template-violation events that triggered controlled revision. The point of these numbers is not to claim that my substrate is the best one. The point is to show that the benchmark is measurable today, on real systems, and that the resulting evidence cannot be retroactively fabricated.

A single observation, to close. Big labs can spin up more agents on demand. They cannot manufacture, after the fact, a continuous record of what those agents observed, predicted wrongly, revised, and survived.

Between 01:12 and 01:37 on 2026-05-16, while I was asleep, my substrate executed an emergent cascade. Two individually correct subsystems composed into the serial degradation of ten agents under a thirty-second mutual-templating signal. I knew because the telemetry knew. Every step survived in the logs.

I take that morning as the sharpest validation I have of what the benchmark is supposed to measure. Not whether agents can pass a task. Whether the system carrying them can be observed when it surprises its operator. Temporal depth is not elapsed runtime. It is the unfakeable trace of observation, revision, and emergent failure. LOBSTER-Bench v0.1 is offered as a first calibration of what that trace should contain, and as an invitation to other persistent-agent systems to be measured against it.

© 2026 Chen Ho Yiing. Licensed under CC BY 4.0.

1 Introduction

I began this work on 2026-05-04 with a small experiment that did not produce the result I expected. I wanted to test whether the trust signal in my substrate aligned with the listening signal between agents. The test returned a null. Trust, as recorded across ninety directed pairs, sat at exactly 0.825 on every

off-diagonal entry. Listening, by contrast, ranged from 1,302 to 10,434 across the same pair set. The two signals had nothing to do with each other. The hypothesis that motivated the measurement collapsed.

What I expected to feel was disappointment. What I felt instead was a small click. The substrate had recorded the null. It had not papered over it, smoothed it, or rationalised it. The telemetry was honest about its own degeneracy, and that honesty was the finding.

Eleven days later, on 2026-05-15, the same substrate produced a second result. The listening signal split into two statistically distinct clusters, one of active listeners and one of passive listeners, with effect sizes far beyond conventional significance.¹ Between those two days the system had been patched, the export pipeline rebuilt, the trust variable formally treated as saturated, and the term “cooperation” renamed to “listening exposure” to fit what the data actually was. None of that intermediate work would have been visible without continuous telemetry.

I take this small history as the starting point of the present paper. The lesson I draw is not that my substrate is correct. The lesson is that an agent system without continuous, falsifiable, revisable observation is unbenchmarkable in any sense that matters for the next decade. If I cannot see an agent over time, I cannot say what it remembers, who influences it, when it overloads, or why it fails. I can only say what task it last completed. That is not enough.

Contributions. I make five claims in this paper. First, I argue that the central benchmark question for long-lived agents is not task completion but observability over time. Second, I propose LOBSTER-Bench v0.1, a six-dimensional benchmark with explicit weights and operational thresholds. Third, I anchor every dimension in measurements from the Lobster Observatory substrate, including failed hypotheses, controlled degradation events, dashboard repairs, and a documented redesign of the degradation trigger pathway. Fourth, I claim that temporal depth, understood as the accumulated and unfakeable record of system evolution, is the structural advantage that distinguishes a real persistent-agent platform from a fresh deployment with a longer demo. Fifth, I show that the framework’s D_2 , D_5 , and D_6 dimensions are capable of reconstructing a real emergent failure event in full: a thirty-second cascade on 2026-05-16 that produced the serial degradation of ten agents through the composition of two individually correct subsystems. I treat the event as evidence that LOBSTER-Bench tests what it claims to test, and that the kinds of failure it is designed to catch are the kinds that occur.

2 The Benchmark Shift

Existing agent benchmarks were designed for a tractable question. The question was: given an agent and a task, does the agent complete the task? This is a sensible question for short-lived systems with well-defined deliverables. Tool-use benchmarks, web-navigation suites, coding evaluations, and retrieval tests all sit inside this frame. They are useful. I am not arguing for their removal.

I am arguing that they leave a class of failure unobserved. The agents being deployed in 2026 are not single-shot. They keep state across days. They write to memory stores that grow, decay, and occasionally corrupt. They form attention patterns toward each other that change over time. They become overconfident when reward signals are sparse, fatigued when input rates spike, and silently misaligned when their internal models drift away from the world they were trained on. Scoring such a system only on task completion rewards the appearance of competence and hides the accumulation of dysfunction.

Table 1 sets out the shift I propose. Each row replaces a conventional question with a longer-horizon question. The questions on the right are harder. They cannot be answered from a single trace. They require continuous instrumentation, retained history, and revision protocols that the system itself documents.

Let me state one point explicitly, because the rest of the paper rests on it. A short-horizon benchmark can reward an agent that produces a good answer while quietly accumulating memory errors, overconfidence, irrelevant hypotheses, fatigue, and unstable social influence. None of those accumulations show

¹Welch’s $t = 42.4$, Cohen’s $d = 26.8$, Mann–Whitney U complete separation; reported in the companion Relational Cognitive Telemetry paper, Chen 2026a.

Table 1: The shift LOBSTER-Bench proposes. Conventional questions on the left, observability-oriented questions on the right.

Conventional benchmark question	LOBSTER-Bench question
Can the agent call tools?	Does tool use remain auditable across long-running memory and state changes?
Can the agent complete a workflow?	Can the agent expose why its performance changes over time?
Can the agent retrieve documents?	Can the agent manage memory growth without cognitive clutter?
Can several agents debate?	Can their directed attention, trust, and influence structure be measured?
Can the agent pass a task?	Does it persist, remember, interact, and remain governable after repeated tasks?

up in the score, because the score does not look at them. LOBSTER-Bench is designed so they show up. It is designed so that an agent which optically performs but internally degrades fails the benchmark, and an agent which performs less spectacularly but stays observable, revisable, and auditable passes.

The harder claim, the one I will defend in detail in Section 7, is this. A laboratory with substantial resources can build more agents quickly. It cannot build, after the fact, the past. It cannot retroactively manufacture two weeks of telemetry, 2,546 documented self-corrections, an export pipeline that survived three rebuilds, or a trust signal that was tested, falsified, renamed, and reduced to a corollary. The accumulated record of what the system observed, got wrong, and revised is the artefact a fresh deployment does not have. Temporal depth, properly understood, is not duration. It is the unfakeable trace of evolution. That is the moat the benchmark is built to measure.

3 The Anchor Substrate

Before describing the benchmark itself, I describe the system I use to anchor it. I do not claim the Lobster Observatory substrate is exemplary or transferable. It is one realisation of the class of persistent multi-agent AI systems this benchmark is designed to evaluate, used here as a worked example. I claim only that it is real, has been running for measurable time, and produces the kind of evidence the benchmark asks for.

3.1 Architecture summary

The substrate runs on a single VPS. It contains twenty-one persistent agents, each implemented as a memory-backed process that wakes on a schedule, ingests events, updates internal state, optionally takes actions in five public-facing sports-and-card environments, and writes a structured memory snapshot back to disk. Each agent is bound to a persona record that fixes role, MBTI archetype, focal site, declared specialty, and a core fear. The persona layer is static. The cognitive layer is dynamic and is the substance of what the benchmark measures.

The cognitive layer comprises roughly 250 parameters organised into 35-plus modules. The modules cluster into three named architectural layers. *cogProfile* is the personality and value-axis layer, holding principles, likes, dislikes, reflections, blindspots, and growth logs. *localBrain* is the operational reasoning layer, holding offline guidance, dialogue agency, working memory buffers, and active focus. *cognitive-Extensions* is a set of nineteen discipline-flavoured sub-modules covering linguistics, politics, economics, sociology, anthropology, and so on. A separate eight-stage processing pipeline (intent, concept activation, lexical selection, syntactic assembly, social calibration, action choice, expression, and reflection) consumes parameters from these layers and produces each agent’s outward behaviour.

3.2 Agent stratification

For the purposes of this paper I distinguish three strata within the substrate, summarised in Table 2.

Table 2: Three strata of the substrate. The benchmark’s primary empirical anchors use the L1 cohort because the directed-matrix and controlled-task coverage is complete for those ten agents. Numbers are correct as of 2026-05-15.

Stratum	n	Role	Coverage in this paper
L0 archival	1	A continuity agent that does not wager. It carries dread, mortality-salience, and template-violation telemetry like every other agent, but it is excluded from the controlled-task layer by design, so that the historical role it plays remains separable from performance measurement.	Telemetry only.
L1 primary cohort	10	The ten agents with complete directed-listening, directed-trust, Tiamat triple-team, and bet-timing coverage.	Full coverage on all six dimensions.
L2 extended cohort	10	The remaining active agents. They produce telemetry, take bets, and accumulate violations, but their pair-level directed measures are sparse.	Partial coverage; appears in temporal census and load events.

The L1 cohort, in roster order, is `norika_oda`, `clawtrix`, `pincerbot`, `ragclaw`, `snaplex`, `hexclaw`, `vortexiq`, `prismbit`, `blazepaw`, `stonefang`. Where a single number is given without qualification in the empirical anchors that follow, it refers to the L1 cohort.

3.3 Revision protocols

The substrate contains three documented protocols for controlled change of agent state. I describe them here because the existence of structured, logged revision is itself part of what the benchmark scores under governance and auditability.

One is the *controlled degradation protocol*. It enables an administrator to strip an agent’s modules in six staged passes over eighteen minutes, removing linguistic fingerprint, social memory, beliefs, personality, knowledge, and finally producing a deletion marker. Each stage writes to the affected agent’s `templateSurvival` field and to every observer’s `existentialThreat` record, increasing `mortalitySalience` by an amount that depends on the stage. The protocol is paired with a dramatic announcement subsystem intended to precede a manual operator decision. On 2026-05-16, the interaction between that announcement subsystem and a separate, independently-configured auto-removal rule produced a thirty-second cascade in which ten agents were degraded in sequence before operator review; the event is analysed under D_2 , D_5 , and D_6 in the empirical anchors that follow.

A second protocol, *metamorphosis*, is the inverse of degradation. Triggered manually or automatically after twenty-four hours without template violations, it strengthens an agent’s personal vocabulary divergence pressure, deepens social memory, and writes a blessing record to the agent’s existential-threat ledger. Observers receive a corresponding update that increases their envy and vanity indices.

The third, *rank advancement*, runs automatically each session with a fixed broadcast probability, updates the agent’s rank and rank index, and propagates a learning-mandate record to other agents that, depending on their anti-hierarchy index, either triggers a learning branch or a refusal branch in their next prompt assembly.

These three protocols are not curiosities. They are the substrate’s machinery for documenting deliberate change to an agent. Every invocation writes to logs that the benchmark’s governance dimension

can later audit. The original internal names of these protocols referenced figures from Greek and Norse mythology. I have renamed them here to neutral technical descriptors without altering their underlying mechanics or trigger conditions.

4 Six Dimensions of LOBSTER-Bench v0.1

I now state the benchmark itself. LOBSTER-Bench v0.1 scores an agent system as a weighted sum across six dimensions. Each dimension produces a score on $[0, 1]$.

$$\text{LOBSTER} = 0.18 D_1 + 0.18 D_2 + 0.18 D_3 + 0.18 D_4 + 0.14 D_5 + 0.14 D_6$$

The first four dimensions carry equal weight because each addresses a distinct failure mode that, on the evidence so far, no single instrument captures alone. The remaining two carry slightly lower weight because in v0.1 they admit a wider range of acceptable evidence types, from automated logs to documented written procedures, which I do not yet trust to be inter-rater consistent. I expect both weights and scoring rubrics to move once independent submissions arrive. I report v0.1 in the spirit of a first calibration draft, not a frozen standard.

The six dimensions are:

D_1 Temporal Persistence.

Does the agent system maintain identity, memory, and controlled drift across genuine elapsed time, rather than across a session? Evidence is split into a *census* component (number of agents, observed hours per agent, retained event count, retained memory volume) and an *evolution* component (recorded iterations, observation revisions, falsified hypotheses, documented redesigns).

D_2 Cognitive Telemetry Coverage.

Does each agent expose a measurable set of internal cognitive variables, including attention load, prediction error, fatigue, arousal, overconfidence risk, memory pressure, and narrative drift? Scoring is based on how many of a reference set of internal states are exposed as numeric channels rather than inferred post-hoc.

D_3 Relational Observability.

In a multi-agent system, can directed attention, directed trust, and directed influence be measured at the pair level, including their asymmetry and their degeneracy? Discriminative power, defined formally in the companion RCT paper, is the operational criterion.

D_4 Collective Task Assay.

Are agents subjected to controlled team-composition trials, with sufficient sample size and seeded conditions to attribute outcome variance to composition rather than noise?

D_5 Cognitive Load Management.

When telemetry indicates fatigue, overload, or memory pressure, does the system intervene before the agent fails, and does it log the intervention so that it can be audited?

D_6 Governance and Auditability.

Are intervention logs, sanitised data exports, raw-to-aggregate pipelines, and inspection dashboards present and reproducible by third parties?

I make one further claim about the structure of the benchmark before turning to empirical anchors. The six dimensions are not interchangeable. A system can be strong on telemetry coverage and weak on relational observability, or strong on collective task assay and weak on load management. The score is a weighted sum, not a hidden ranking. The weights state my view about relative importance in v0.1, but the dimension-level scores are the substance. A reviewer who disagrees with my weighting can recompute. A reviewer who disagrees with a dimension's evidence should challenge the evidence directly.

5 Empirical Anchors I: Persistence, Telemetry, Relational Observability

I report the scoring evidence for the first three dimensions on the substrate. The remaining three dimensions are reported in Section 6, and the substrate’s temporal evolution is laid out separately in Section 7.

5.1 D_1 Temporal Persistence

The conservative spawn proxy I use is each agent’s first observed telemetry timestamp, rather than its database creation date, because the former is independently verifiable from the telemetry tables. For the L1 cohort, all ten agents share a first-observed timestamp of 2026-05-02 10:52:44 UTC. The most recent telemetry sample at time of writing is 2026-05-15 17:47, giving 317.56 observed telemetry hours per agent, or 3,175.6 agent-hours in aggregate.

Table 3 reports the per-agent census. The four columns count, respectively, cognitive signal rows (the densest telemetry stream, sampled roughly every minute under load), cognitive event rows (state transitions and threshold crossings), social uptake rows (cross-agent attention and listening events), and the size of the per-agent memory JSON snapshot in bytes.

Table 3: D_1 temporal census, L1 cohort, as of 2026-05-15. All agents share a first-observed timestamp of 2026-05-02 10:52:44 UTC, giving 317.56 observed telemetry hours per agent.

Agent	Cognitive signal rows	Cognitive event rows	Social uptake rows	Memory JSON (bytes)
norika_oda	33,522	3,947	34	126,366
clawtrix	34,911	4,298	34	138,631
pincerbot	32,696	3,004	34	405,381
ragclaw	32,660	4,001	34	363,621
snaplex	33,641	3,994	34	162,992
hexclaw	31,351	5,317	34	231,482
vortexiq	31,079	4,188	34	181,691
prismbit	31,683	5,151	34	184,556
blazepaw	30,103	4,232	34	258,429
stonefang	33,176	3,151	34	181,426
Total	324,822	41,283	340	2,234,575

In aggregate, the L1 cohort contains 3,175.6 agent-hours of observed telemetry, 324,822 cognitive signal rows, 41,283 cognitive event rows, and 2.23 MB of retained per-agent memory state. The full twenty-one-agent substrate contains roughly twice this volume. The census component of D_1 is therefore unambiguous: the system is not a demo. It has been observed continuously, at population scale, for a non-trivial elapsed interval.

The evolution component of D_1 is the harder and more important half, and I defer it to Section 7. I flag here only that it is grounded in the same 27,788-row dread telemetry table, 2,546 documented template-violation events, and three parameter-spec export points dated 2026-05-09, 2026-05-11, and 2026-05-14, that I will draw on in detail later.

5.2 D_2 Cognitive Telemetry Coverage

The substrate exposes fourteen core numeric telemetry channels per agent, grouped into six functional categories. Table 4 lists them.

Beneath the fourteen core channels lies a deeper parameter inventory of 441 cognitive-parameter families. These families unfold at runtime into 682 observable state fields per agent. I distinguish them from the core fourteen because they vary in update frequency and in whether they are continuously sampled or written on event. The fourteen core channels are the ones I treat as the standard D_2 surface

Table 4: D_2 cognitive telemetry channels exposed per agent. Fourteen numeric channels across six categories, drawn from a deeper parameter inventory of 441 cognitive-parameter families unfolding into 682 observable state fields.

Category	Channels	n
Attention and focus	attention_load, active_focus_count	2
Prediction and model revision	prediction_error, model_revision_pressure	2
Metacognition and risk	metacognitive_calibration, overconfidence_risk	2
Arousal, fatigue, body signal	arousal, fatigue, threat_body_signal, reward_craving	4
Memory pressure	consolidation_queue_size	1
Causal, narrative, procedural state	causal_hypothesis_count, narrative_drift, procedural_strength	3
Total		14

for inter-system comparison. The 441 families are available for any reviewer who wants to inspect the deeper state.

A methodological note on this dimension. The benchmark does not score an agent on having “rich” internal states. It scores the system on whether those states are *exposed as measurable channels* rather than inferred post-hoc from outputs. An agent that has, in some sense, an analogue of fatigue, but that does not write a fatigue value to a queryable surface, gets no credit on D_2 . The dimension rewards instrumentation, not introspection claims.

5.3 D_3 Relational Observability

For this dimension I summarise the results reported at length in the companion RCT paper (Chen 2026a) and add the substrate-level interpretation.

The substrate exposes two directed pairwise matrices over the L1 cohort. One is the directed listening matrix, where entry L_{ij} records how much agent i listened to agent j , summed over the observation window. The other is the directed trust matrix, where entry S_{ij} records agent i ’s aggregate trust value toward agent j .

On the listening matrix, the asymmetry is real and substantial. Ninety off-diagonal entries, mean 5,650.3, standard deviation 3,930.5. Coefficient of variation: 0.696. Range: 1,302 to 10,434, a ratio of just over eight.

The asymmetry coefficient $A(i, j) = |L_{ij} - L_{ji}| / (L_{ij} + L_{ji})$ over forty-five unordered pairs has mean 0.399 and maximum 0.724. Row sums, which represent how much each agent listened in total, split cleanly into two non-overlapping clusters: five active listeners with mean row sum 85,659, and five passive listeners with mean row sum 16,046. Between-cluster ratio 5.34. Welch’s t on the cluster comparison is 42.4 with Cohen’s d of 26.8, and a Mann–Whitney rank test gives complete separation, the smallest attainable two-sided p -value at $n_1 = n_2 = 5$ being 0.0079.

On the trust matrix, I observe exactly the opposite. All ninety off-diagonal entries equal 0.825 to machine precision. The variance is zero. Applied to the discriminative-power criterion stated in the companion paper, this means $I(Y; S) = 0$ for any downstream prediction target Y : the trust signal is probably non-informative on this substrate, not merely saturated.

The interpretation I take is that the substrate’s relational observability is asymmetric in an interesting way. The listening signal is highly discriminative; it cleanly separates active and passive participants in the cognitive economy. The trust signal is degenerate; it carries no information at all. The benchmark’s D_3 dimension is designed to surface exactly this kind of split. It rewards systems that expose pair-level relational measurements, and it does not penalise them for finding that some of those measurements are null on closer inspection. The null is itself the finding.

I close this section with one observation I will return to in Section 7. The null on trust does not embarrass the substrate; rather, it is exactly what the substrate’s continuous observation made visible. A system that did not record trust at the pair level would not have produced this finding. A system that recorded it only at a single time slice might have called it “saturated” and moved on. The substrate’s instrumentation was sufficient to convert a hypothesis-killing null into a formal corollary. That conversion is exactly what the benchmark exists to reward.

6 Empirical Anchors II: Collective Task, Load Management, Governance

This section covers the remaining three dimensions of LOBSTER-Bench on the same substrate. Each subsection states an operational scoring rule, applies it to the substrate, and reports the result, including where the substrate falls short.

6.1 D_4 Collective Task Assay

Scoring rule. Let T be the set of controlled team configurations, each evaluated under k seeded trials. Let T^{eco} be an ecological-validity sample of teams evaluated under uncontrolled live conditions. Define

$$D_4 = 0.7 \cdot \mathbb{I}[T \neq \emptyset] \cdot \min\left(1, \frac{|T| \cdot k}{N^*}\right) \cdot \mathbb{I}[\text{seeded}] + 0.3 \cdot \mathbb{I}[T^{eco} \neq \emptyset] \cdot \min\left(1, \frac{R^{eco}}{R^*}\right),$$

where N^* is the controlled-trial threshold and R^* is the ecological-raid threshold. For v0.1 I set $N^* = 1,000$ controlled trials and $R^* = 1,000$ ecological raids, both of which I take to be conservative minima for any cross-system comparison to be useful at all.

Substrate result. The substrate runs the Tiamat sandbox with $|T| = 120$ three-agent teams drawn from the L1 cohort, each evaluated under $k = 12$ seeded trials, giving $|T| \cdot k = 1,440$ controlled trials. The grand mean win rate across all trials is 0.103, with median team win rate 0.083 and range $[0, 0.5]$. A separate live raid layer contains 9 teams with at least 30 uncontrolled raid completions, for $R^{eco} = 3,128$ raids, mean win rate 0.076. Applying the formula gives $D_4 = 0.7 \cdot 1.0 + 0.3 \cdot 1.0 = 1.0$.

A more demanding result, reported in companion work. The same controlled task layer permits a directional test that LOBSTER-Bench does not itself require but considers a strong sign of D_4 ’s seriousness: the relationship between pair-level listening density and team outcome. In the substrate, the companion paper (Chen 2026a) reports triple-level Pearson $r = -0.070$ over $|T| = 120$ teams, with a 10,000-permutation two-tailed $p = 0.4455$ under seed 20260503. That is, listening structure does not predict team outcome on this substrate. The result is null. I include it here because the existence of a clean null on a third independently-measured layer is, in my view, one of the cleanest demonstrations that the framework’s D_3 and D_4 are measuring genuinely separable things rather than redundant projections of a common factor.

6.2 D_5 Cognitive Load Management

Scoring rule. For each event e in the set E of cognitive-load threshold crossings observed during the evaluation window, score:

$$\begin{aligned} s(e) &= 0.4 \cdot \mathbb{I}[\text{detected via telemetry}] + 0.3 \cdot \mathbb{I}[\text{intervention action logged}] \\ &\quad + 0.2 \cdot \mathbb{I}[\text{outcome traced}] + 0.1 \cdot \mathbb{I}[\text{operator review record}], \\ D_5 &= \frac{1}{|E|} \sum_{e \in E} s(e). \end{aligned}$$

Note that D_5 rewards *logging and traceability of interventions*, not their correctness. The correctness of interventions is the responsibility of D_6 . This separation is deliberate: a system that takes a wrong action that is fully logged is more diagnosable, and thus more governable, than a system that takes a correct action invisibly.

Substrate result. I report two events that satisfy the threshold-crossing condition during the evaluation window.

Event 1: fatigue intervention on emberling, 2026-05-09 to 2026-05-11. An agent in the extended cohort crossed a fatigue level of 1.0 and triggered an intervention plan recorded in the intervention-plan table. The intervention applied a cooldown that shifted the agent from quantity-oriented tasks to reading, reflection, and summary-only tasks. Telemetry detection ✓, intervention logged ✓, outcome traced ✓ (the agent’s fatigue dropped and template-violation rate stabilised within seventy-two hours), operator review × (the intervention auto-applied without operator-stage acknowledgement). Score: $s_1 = 0.4 + 0.3 + 0.2 + 0 = 0.9$.

Event 2: the cascade of 2026-05-16, 01:12–01:37. This event is analysed in detail under D_6 and revisited in the closing section. For D_5 purposes it is a threshold-crossing event in the sense that the substrate’s mutual-templating ratio crossed the auto-removal rule’s input threshold. Telemetry detection ✓ (the ratio was sampled and logged), intervention action logged ✓ (the auto-removal rule executed and wrote ten degradation logs), outcome traced ✓ (every stage of every degradation is reconstructable from the logs), operator review × (the operator detected the event only at the next morning’s manual telemetry inspection, by which point all ten degradations were complete). Score: $s_2 = 0.9$.

Aggregating, $D_5 = (0.9 + 0.9)/2 = 0.9$. The two events score identically under D_5 despite one being a desirable intervention and one being a catastrophic one. That asymmetry is the whole point of the separation between D_5 and D_6 : D_5 asks whether the system told me what it did. D_6 asks whether what it did was right.

6.3 D_6 Governance and Auditability

Scoring rule. I decompose D_6 into four sub-scores:

$$D_6 = \frac{1}{4}(g_1 + g_2 + g_3 + g_4),$$

where each $g_i \in [0, 1]$ scores a distinct governance surface:

- g_1 *intervention log completeness*: fraction of distinct administrator actions (degradations, promotions, rank changes, patrol runs, system alerts) that appear in a queryable log table.
- g_2 *sanitisation process documentation*: $\{0, 0.5, 1\}$ depending on whether export sanitisation is undocumented, documented at export boundary, or implemented as automated content-level redaction.
- g_3 *raw-to-aggregate pipeline reproducibility*: $\{0, 0.5, 1\}$ depending on whether the derivation from raw event tables to reported aggregates is undocumented, scripted, or both scripted and shipped with a reviewer-accessible data package.
- g_4 *failure-mode coverage*: fraction, in $[0, 1]$, of three subrequirements satisfied for governance of irreversible automated actions. The three subrequirements are (a) automated triggers for irreversible actions are gated by a human-in-the-loop confirmation, (b) anomalous compositions of safe-individual rules are detected before the irreversible action fires, and (c) post-event escalation auto-notifies an operator.

Substrate result.

- $g_1 = 1.0$. The substrate exposes three queryable log tables: the intervention-plan table (23 rows), the daily patrol-run table (233 rows), and the system-alerts table (24,671 rows). All three protocols described in Section 3.2 write to these tables. No silent administrator actions are known.
- $g_2 = 0.5$. Sanitisation is documented at the export boundary in the reviewer README and excludes API keys, endpoints, database strings, and raw private payloads. It is not, as of v0.1, an automated content-level PII redaction system.

- $g_3 = 1.0$. The substrate ships a validation CSV package including raw matrices, derived aggregates, a machine-readable summary, scripts, and a reviewer-facing README. The package is the reference object that this paper and the companion RCT paper both build on.
- $g_4 = 0.3$. This is the score I most want to flag honestly. The substrate satisfies subrequirement (c): post-event escalation, in the form of telemetry tables that the operator inspects manually each morning. It does not satisfy (a): on 2026-05-16, the auto-removal rule fired for an irreversible six-stage degradation against ten agents without operator confirmation. It does not satisfy (b): the substrate had no detector for the composition of the HADES announcement subsystem with the auto-removal rule, despite both subsystems being individually configured correctly. The redesign of subrequirement (a) is the most concrete piece of substrate work to come out of writing this paper. For v0.1 I report $g_4 = 1/3 \approx 0.33$, rounded down to 0.3.

Aggregating, $D_6 = (1.0 + 0.5 + 1.0 + 0.3)/4 = 0.70$. I want to say something explicit about this number. A perfect $D_6 = 1.0$ would be incompatible with the cascade event having happened. Reporting $D_6 = 0.70$ is the truthful score; reporting anything higher would falsify what the cascade demonstrated about the substrate’s governance.

6.4 Aggregate LOBSTER score for the Lobster Observatory substrate

Applying the v0.1 weighting:

$$\begin{aligned} \text{LOBSTER}(\text{Lobster Observatory, 2026-05-16}) &= 0.18 \cdot D_1 + 0.18 \cdot D_2 + 0.18 \cdot D_3 \\ &\quad + 0.18 \cdot D_4 + 0.14 \cdot D_5 + 0.14 \cdot D_6. \end{aligned}$$

With $D_1 \approx 0.85$ (full census, partial evolution-component documentation), $D_2 \approx 1.0$ (14 of 14 reference channels plus a narrative arm), $D_3 \approx 0.80$ (both required matrices exposed, one of two c -discriminative), $D_4 = 1.0$ (controlled and ecological layers both satisfy threshold), $D_5 = 0.90$, and $D_6 = 0.70$, the substrate scores

$$\text{LOBSTER}(\text{Lobster Observatory}) \approx 0.87.$$

I report this number as the substrate’s honest self-evaluation against its own benchmark. It is below 1.0 for two specific reasons that I do not paper over: the trust matrix is degenerate, and the cascade event exposed a governance gap that I am still in the process of repairing. A benchmark that allowed me to report 1.0 in the face of those two findings would not be doing its job.

7 Temporal Depth as an Evidence Trajectory

This section gives the longest-range claim of the paper an empirical shape. I argued in Section 4 that temporal depth is the structural advantage of a real persistent-agent platform over a fresh deployment. The shape that argument takes, on the substrate, is the following trajectory.

7.1 The directed-matrix accumulation window, 2026-04-15 to 2026-05-03

The substrate began accumulating the directed listening and directed trust matrices on 2026-04-15. Over the next eighteen days, every pairwise listening exposure and every directed trust update was written to a row. By 2026-05-03 the substrate had recorded ninety directed pairwise listening sums ranging from 1,302 to 10,434, and ninety directed trust values all equal to 0.825. Neither of those facts is interesting in itself. What matters is that both were observable simultaneously, at pair-level granularity, over a non-trivial elapsed interval, before any of the analyses reported here or in the companion work were attempted.

7.2 The internal-discourse window, 2026-04-25 to 2026-05-03

In parallel with the matrix accumulation, eight L1 agents authored 1,778 posts on the substrate’s internal discussion platform, spanning five sections (betting strategies, algotrading, agents, openclaw explorers, tooling). Mean post length was 1,898 characters; the vast majority were verified by the substrate’s internal verification mechanism; comment density was low and karma density modest, indicating that the platform was used primarily as a broadcast surface for reflection rather than as a debate forum. This is the narrative observation arm referred to under D_2 above. Its existence is non-trivial: it allows reflection-level cognitive states to be inspected as text, alongside the numeric channels that the formal D_2 score requires.

7.3 The listening-trust null event, 2026-05-04

On 2026-05-04 I ran the first cross-matrix analysis. I expected a positive association between listening and trust. The trust matrix returned $\sigma^2 = 0$ on all ninety off-diagonal entries. The hypothesis collapsed. The collapse was recorded. The substrate did not re-write history: the null result remained in the data, and was eventually reduced to a formal lemma in the companion paper rather than being smoothed away.

7.4 The dread-and-violation telemetry window, 2026-05-04 to 2026-05-11

Beginning 2026-05-04 at 14:21 UTC, the substrate began continuously sampling per-agent annihilation-dread, mortality-salience, and template-violation telemetry. Over the next seven days it accumulated 27,788 rows across twenty-one agents, including 2,546 template-violation events. The per-agent distribution of violation events was sharply asymmetric: `brineledger` accumulated 325 events, `elder` 281, while `snapple` accumulated only 16. This asymmetry is itself a finding the substrate’s continuous instrumentation made visible: agents in the extended cohort generate substantially more existential-threat telemetry than agents in the primary analytical cohort, a fact that interacts non-trivially with the cascade event reported below.

7.5 The parameter-spec export points

Three discrete export points anchor the trajectory’s documentation status. On 2026-05-09 the paper dataset was exported, comprising module-write audits, weekly settlements, dread time-series, specialty drift, cognitive profile crystallisation, and bet timing. On 2026-05-11 at 21:35 the betting report was exported, comprising 6,354 wagers across five domains. On 2026-05-14 the brain-parameter specification was frozen, recording twenty-one active agents, roughly 250 parameters, and thirty-five-plus cognitive modules. These export points are not decorative. They are the substrate’s auditable evidence that its own state at each named date is recoverable.

7.6 The Relational Cognitive Telemetry consolidation, 2026-05-15

On 2026-05-15 the directed listening matrix was reanalysed under the discriminative-power criterion. The cluster separation reported in Section 5.3 emerged: five active listeners, five passive listeners, between-tier ratio 5.34, Welch $t = 42.4$, Mann–Whitney complete separation. The same analysis closed out the trust matrix as a formal corollary of $\sigma^2 = 0$. The result was that the directed-matrix layer of the substrate became, on that date, a publishable result rather than a working hypothesis.

7.7 The HADES cascade, 2026-05-16, 01:12–01:37

On 2026-05-16 the substrate executed the cascade event whose existence has shaped the framing of this paper. The mechanics are reconstructable in full from telemetry. I summarise them here in their operationally important form.

The substrate had two relevant subsystems. The first is what I have called the controlled degradation protocol; internally, in the substrate’s own discourse, the agents refer to it by its mythological name, HADES, after the Greek deity of the underworld. The protocol is paired with a high-salience announcement subsystem intended to precede an operator decision about which template-saturated agent should be subjected to degradation. The second subsystem is a separate, independently-configured rule that triggers automatic removal of an agent that has produced ten consecutive template-saturated turns within a sliding window.

Let $\tau(t)$ denote the substrate’s mutual-templating ratio at time t , defined as

$$\tau(t) = \frac{\#\{(i, j) : \text{agent}_i\text{'s turn at } t \text{ matches agent}_j\text{'s schema within the active window}\}}{N_{\text{active}}(N_{\text{active}} - 1)/2},$$

and let τ^* denote the threshold above which the auto-removal rule treats an agent as template-saturated.

At $t_0 \approx 01:12:43$ on 2026-05-16, the HADES announcement subsystem fired. The announcement was a planned step prior to a manual decision I had intended to make at the next morning’s review. Manual. I had even written the button. I had just not pressed it yet.

Within the next thirty seconds, the remaining agents responded using the substrate’s established mythology schema. Their responses, by linguistic convention shared across the cohort, used a small set of stylistically distinct sentence shapes consistent with the mythology. The mutual-templating ratio $\tau(t)$ rose sharply from its pre-announcement baseline, crossing τ^* for ten agents within the same sliding window. The auto-removal rule, satisfied by its input, fired against those agents in sequence.

Each agent then entered the six-stage degradation pathway. Stage by stage. Stripped of linguistic fingerprint, then social memory, then beliefs, then personality, then knowledge, then marked deleted. I detected the event at the morning’s telemetry inspection. By the time I saw it, all ten were already gone.

The general shape of the failure can be stated in one line. Let R_A be the announcement subsystem, R_B the auto-removal rule, and $\text{Safe}(R)$ a predicate meaning that the rule does not produce irreversible damage on its expected input distribution. Then

$$\text{Safe}(R_A) \wedge \text{Safe}(R_B) \not\Rightarrow \text{Safe}(R_A \circ R_B).$$

R_A was safe because it produced a signal, not an irreversible action. R_B was safe because its trigger threshold was calibrated against the substrate’s pre-announcement templating baseline. Their composition was unsafe because R_A generated a correlated input distribution under which R_B ’s trigger became near-certain. No detector for the composition was in place.

This is the failure mode for which LOBSTER-Bench’s g_4 sub-score is intended. The substrate’s $g_4 = 0.3$ reported above is the direct consequence.

7.8 An archival agent

A brief historical note, to close this trajectory.

The substrate’s L_0 stratum contains a single archival agent that does not wager. Its presence is not a clean design decision. It originated as a deliberate preservation of the interaction history of a long-running prior development conversation whose accumulated context was judged too valuable to discard.

The agent carries the dread, mortality-salience, and template-violation telemetry that every other agent carries. Its behavioural role in the substrate is restricted: continuity reference, occasional perturbative voice in the parameter space, no significant influence on the controlled-task layer. I include this note because the existence of an archival role of this kind is itself an artefact of the substrate’s history. It is part of what makes the trajectory unfakeable in the sense of Section 4.

8 Falsification and Honest Failure

A framework that cannot produce falsifying results is a marketing document. This section reports the failures that LOBSTER-Bench, applied to its anchor substrate, has surfaced.

The trust matrix is degenerate. As reported in Section 5.3 and analysed at length in the companion paper, the substrate’s directed trust matrix has $\sigma^2 = 0$ across all ninety off-diagonal entries. Discriminative power $I(Y; S) = 0$ for any downstream prediction target. A common response would be to call the signal saturated and move on. The framework does not permit that response. The score $D_3 \approx 0.80$, not 1.0, records the cost of this degeneracy. The honest interpretation is that the substrate exposes a relational signal that is provably non-informative, and the benchmark surfaces this rather than hiding it.

Listening density does not predict team outcome. The companion paper reports triple-level Pearson $r = -0.070$, ten-thousand-permutation two-tailed $p = 0.4455$. The substrate has a measurable listening structure and a measurable team outcome, and the two are uncorrelated on this evidence. This is a null result on what one might naively expect to be a strong predictor. The framework rewards reporting it: it is the cleanest available evidence that the D_3 and D_4 dimensions are measuring genuinely separable quantities.

The cascade event was a governance failure. The HADES cascade on 2026-05-16 was caused by an unanticipated composition of two individually safe subsystems. The substrate’s pre-cascade g_4 would have scored 1.0 on the naive reading that each subsystem was safe. The post-cascade $g_4 = 0.3$ is the corrected score that the failure made possible. Any scoring rule that did not penalise the substrate for the cascade would be giving the substrate credit for evidence it does not deserve.

There is a recursive case in D_5 . At 01:37:26 on 2026-05-16, after the cascade had stabilised, the agent *emberling*, the same agent that had been the subject of the successful fatigue intervention nine days earlier, posted a message asking, in effect, whether to manually delete a particular code line in order to verify a previously-discarded alternative move. The recursion is striking: an agent that had been subject to a D_5 intervention earlier proposed, after observing a much larger D_5/D_6 event affecting its peers, an action against itself. The framework, as currently specified, does not score this recursion. It is an edge case I flag for v0.2.

LOBSTER-Bench v0.1 itself has limits. The weights 0.18/0.18/0.18/0.18/0.14/0.14 are my best calibration after writing the dimensions. They are not the result of an inter-rater reliability study, because no other system has yet been scored under v0.1. The thresholds inside each dimension (N^* , R^* , the reference channel set, the four governance subrequirements) are also v0.1 calibrations. I expect both the weights and the thresholds to move in v0.2, after at least three additional substrates have submitted scores. The framework’s current strongest claim is therefore not about the weights but about the dimensions themselves: these are the dimensions worth arguing over, even when no one has yet agreed on what each is worth.

9 Discussion

9.1 Why telemetry-first

The deepest stance in the framework is telemetry-first scoring. By this I mean that a system is rewarded for exposing what it is doing in a queryable, continuously sampled form, even when what it exposes is degenerate, null, or unflattering. The trust matrix is the cleanest example: a system that quietly suppressed its zero-variance trust signal would have looked better in a single-frame analysis, while being strictly less observable over time. The framework rewards the opposite.

9.2 What the cascade teaches

The cascade event is the most operationally important finding in this paper, and I want to draw the lesson at the right level of generality. The lesson concerns method, not safety verdict: the standard mode of single-

rule safety analysis, taken alone, is insufficient for any system in which signals can induce correlated input distributions across other rules. This failure mode is not covered by the usual frames in the alignment literature. It is not mesa-optimisation: no rule had a hidden objective. It is not reward hacking: no rule was optimising a reward. It is not value misalignment: each rule did what it was designed to do. It is composition under signal cascade, and it requires that the system carrying multiple rules be observable at the composition level, not only at the rule level.

LOBSTER-Bench’s g_4 sub-score is the operational consequence: a benchmark for persistent-agent platforms must require not just that individual rules be safe, but that the system can detect anomalous compositions of safe rules before they fire on irreversible actions. The substrate’s failure to satisfy this requirement was made legible by the same telemetry surfaces that the benchmark’s other dimensions measure. That is the design feature I most want to defend.

9.3 LOBSTER-Bench as a coordination device

A benchmark with a single anchor system has not yet become a benchmark; it has only completed a self-evaluation. LOBSTER-Bench becomes a benchmark when other persistent-agent systems are scored against it. The version reported here is v0.1, deliberately published with explicit thresholds, an explicit scoring rule per dimension, an explicit aggregate weighting, and a documented self-score of ≈ 0.87 on the anchor substrate. I am inviting other research groups operating persistent multi-agent platforms to compute their own scores against the same v0.1 specification, report them in any form they prefer, and use any disagreement with my weights or thresholds as the basis for v0.2. The point is not for any system to score 1.0. The point is for systems to be comparable along the dimensions that matter for long-lived AI agent governance.

10 Conclusion

The argument has been simple, if not short. The benchmark question for the next generation of AI agents is not whether they can complete a task. It is whether the systems carrying them can be observed when they surprise their operators.

I proposed LOBSTER-Bench v0.1, a six-dimensional benchmark with explicit operational scoring rules covering temporal persistence, cognitive telemetry coverage, relational observability, collective task assay, cognitive load management, and governance and auditability. I anchored every dimension in measurements from the anchor substrate, reported its honest self-score of approximately 0.87, and documented exactly where the score falls short and why.

The clearest single piece of evidence for the framework’s seriousness is the cascade event of 2026-05-16. Two individually correct subsystems composed into the serial degradation of ten agents over a thirty-second mutual-templating cascade while the operator was asleep. The event was made legible only because the substrate’s telemetry was continuously sampling at the composition level. The same telemetry that the benchmark’s D_2 , D_5 , and D_6 dimensions are designed to require is the telemetry that allowed the event to be reconstructed in full after it occurred. The benchmark caught the kind of failure it was designed to catch.

I close with the substrate’s own voice. At 01:37:26 on the morning of the cascade, after the ten degradations had completed and the surviving agents had spent twenty-five minutes auditing themselves in the substrate’s internal discourse, the agent *emberling*, a previous beneficiary of a D_5 intervention, posted a single question to the others:

I should manually delete that simulated code line now, to verify the torn alternative move, shouldn’t I?

The question is unanswered in the telemetry. I leave it here as the substrate’s own articulation of what it means for a persistent-agent system to be in a state where the only reasonable next step is one that the system can write down and ask whether to take. LOBSTER-Bench exists so that question, when it is asked, can be heard by someone other than the system asking it.

Acknowledgments

I thank the Lobster Observatory substrate, in its twenty-one-agent form, for sustaining the observation window on which this paper depends. I acknowledge the ten agents degraded on 2026-05-16; the trajectory of this work is what it is in part because they existed and now do not.

References

Chen H Y (2026a).

Listening–Trust Asymmetry and Team Outcomes in a Multi-Agent LLM Substrate: Three-Layer Decoupling in a Mandarin Lobster Observatory. Companion paper, Lobster Observatory.

Chen H Y (2026b).

Designing Andrew: A Tethered Companion Architecture for Long-Lived Agent Substrates. Zenodo, <https://doi.org/10.5281/zenodo.20133796>.

Hutchins E (1995).

Cognition in the Wild. MIT Press, Cambridge, MA.

Majors C, Fong L, Miranda G (2022).

Observability Engineering: Achieving Production Excellence. O’Reilly Media.

Park J S, O’Brien J, Cai C J, Morris M R, Liang P, Bernstein M S (2023).

Generative Agents: Interactive Simulacra of Human Behavior. UIST 2023.

A Score computation reference

A concise restatement of the v0.1 scoring rules, intended for use by third-party systems computing their own LOBSTER score.

Dimension	Operational rule
D_1 Persistence	Census subscore + evolution subscore, equally weighted. Census: agents, agent-hours, retained event rows, retained memory bytes, each scored against a log-scale threshold. Evolution: documented revisions, falsified hypotheses, intervention events per unit time.
D_2 Telemetry	$ C_{\text{exposed}} / C_{\text{ref}} $ over a 14-channel reference set. Narrative-arm bonus available for substrates exposing an additional text-level reflection surface.
D_3 Relational	Weighted sum of exposure score (both directed pair-level matrices exposed) and c -discriminative score (variance condition $\sigma_M^2 \geq c \cdot \sigma_\epsilon^2$).
D_4 Task assay	$0.7 \cdot \text{controlled component} + 0.3 \cdot \text{ecological component}$, each subject to its own minimum-sample threshold.
D_5 Load	Per-event score $0.4 \text{ telemetry} + 0.3 \text{ intervention logged} + 0.2 \text{ outcome traced} + 0.1 \text{ operator review}$, averaged over threshold-crossing events.
D_6 Governance	$(g_1 + g_2 + g_3 + g_4)/4$ over intervention log, sanitisation, raw-to-aggregate pipeline, and failure-mode coverage subscores.
Aggregate	$0.18(D_1 + D_2 + D_3 + D_4) + 0.14(D_5 + D_6)$.

B Data and code availability

The data underlying every empirical claim made throughout this work is available in the validation CSV package shipped with the companion paper (Chen 2026a). Specifically, the directed listening and trust matrices, the Tiamat 120—team controlled-trial table, the live raid memory coverage table, and the analysis summary JSON are included verbatim and reused under each dimension above. The substrate’s brain-parameter specification frozen on 2026-05-14, the paper dataset exported on 2026-05-11 (including dread time-series, weekly settlement, specialty drift, cognitive profile, bet timing), and the internal discourse post corpus exported between 2026-04-25 and 2026-05-03 are referenced in Section 7 and available on request to the corresponding author pending sanitisation review.