Judgment Preservation in Persistent AI Agents: A Unified Cognitive Substrate for Routing Reinforcement and Metacognitive Continuity

William Kyle Million
IntuiTek[1]

February 2026

Abstract

Persistent AI agents operating across multiple sessions face a form of cognitive degradation that existing memory systems do not address: the loss of *reasoning texture* during context compaction. While current approaches preserve facts, preferences, and episodic records, they do not preserve the qualitative judgment that emerges from sustained engagement—the eliminated hypotheses, the reasoning chains connecting observations, the negative knowledge about what was tried and failed. We identify this as the **judgment gap**: a category of cognitive quality absent from existing agent memory taxonomies. To address it, we present the **Unified Cognitive Substrate (UCS)**, a system that fuses quantitative capability routing reinforcement with qualitative metacognitive preservation into a single deployable framework. UCS combines a toroidal graph engine that learns which capability transitions produce value (and strengthens those paths through reinforcement) with a structured reflection protocol that preserves *why* those transitions work and *when they fail*. A translation layer connects quantitative artifacts (wormholes, attractors, resonances) to human-readable methodology annotations. We validate the system through a three-phase empirical investigation comprising 11 experiments across two engine variants, identifying and resolving an architectural disconnection between the energy surface and routing decisions. The resulting system (v1.2) demonstrates context-sensitive routing with 1,563× improvement in advisory differentiation over the original architecture. The framework is open-source and deploys as a boot-per-call skill for any agent platform. Code and experiments are available at the repository accompanying this paper.

1 Introduction

Large language model (LLM) agents operating in persistent environments face a fundamental challenge that has been well-documented in recent literature: the management of memory across sessions. Context windows are finite, and when they fill, some form of compression occurs—either explicit summarization, eviction of older messages, or retrieval from external stores. A growing body of work addresses this challenge through memory architectures of increasing sophistication, including graph-based memory stores [Mem0, 2025], virtual memory paging systems [Packer et al., 2023], field-theoretic retrieval [Mitra, 2026], and compressed cognitive states [ACC Authors, 2026].

However, these systems share a common assumption: that the primary challenge is *information management*—deciding what to store, how to retrieve it, and when to discard it. We argue that this framing misses a distinct category of cognitive quality that is destroyed by compression and cannot be recovered through retrieval alone.

Consider an agent that has spent 40 messages debugging a complex system. By message 40, the agent's responses are informed by a contextual richness that includes: dozens of eliminated hypotheses, the specific pattern that first triggered suspicion, reasoning chains connecting

seemingly unrelated observations, and implicit knowledge about which approaches don't work in this domain. If this conversation is compacted, the summary might read: "The bug was in module X, caused by race condition Y." A future agent instance reading this summary has the *conclusion* but not the *expertise*. The reasoning texture—the judgment that accumulated through sustained engagement—is lost.

Critically, the agent experiencing compaction *cannot perceive this loss*. The summary feels complete because the agent has no comparison point against the pre-compaction state. This creates what we term the **introspection gap**: the degradation is invisible to the system being degraded.

We call the missing category **judgment**: the qualitative cognitive quality that emerges from experience and is distinct from both factual knowledge and episodic memory. In the comprehensive "Memory in the Age of AI Agents" survey [Hu et al., 2025], which categorizes agent memory by form (token-level, parametric, latent), function (factual, experiential, working), and dynamics (formation, evolution, retrieval), judgment does not appear as a category. We propose that it should.

1.1 Contributions

This paper makes the following contributions:

1. **Identification of the judgment gap.** We articulate a category of cognitive quality—reasoning texture, negative knowledge, methodology—that existing agent memory taxonomies do not track and that is destroyed by standard compaction procedures.

2. **The Unified Cognitive Substrate (UCS).** A framework that fuses two systems: (a) a toroidal graph engine for quantitative capability routing reinforcement and (b) a structured metacognitive preservation protocol for qualitative judgment. A translation layer connects quantitative patterns to human-readable explanations.

3. **Empirical validation.** An 11-experiment investigation comparing the full system against a stripped-down baseline, identifying an architectural disconnection, and validating the fix through controlled ablation.

4. **Open-source implementation.** A deployable, boot-per-call skill with 17 passing tests, suitable for integration with any agent platform.

2 Related Work

2.1 Memory Management for LLM Agents

The dominant approaches to persistent agent memory fall into several categories. **Retrieval-augmented systems** such as Mem0 [Mem0, 2025] store past interactions in vector databases and retrieve relevant fragments by semantic similarity. These systems optimize *what is in the context window* at query time but do not address *how reasoning quality degrades* across sessions.

Virtual memory systems, exemplified by MemGPT/Letta [Packer et al., 2023], treat the context window as working memory and implement OS-style paging between core memory, archival storage, and recall memory. This architecture manages the information flow across memory tiers effectively, but the memory operations are fundamentally token management—they do not model or preserve the cognitive processes that produced those tokens.

Compressed state systems such as Adaptive Context Control (ACC) [ACC Authors, 2026] replace transcript replay with a bounded persistent state variable containing goals, constraints, predictive cues, and uncertainty signals. This overlaps with our externalization protocol but

compresses into a single state variable rather than a structured knowledge architecture with provenance tracking and reopening conditions.

Field-theoretic approaches [Mitra, 2026] treat memories as continuous fields governed by partial differential equations, with diffusion through semantic space and thermodynamic decay. This work shares conceptual lineage with our toroidal energy surface—both apply physics-inspired continuous dynamics to agent cognition. The key distinction is in application domain: Mitra applies field theory to *memory retrieval* (stored memories are located in semantic space); we apply toroidal topology to *capability routing* (capabilities are nodes on a graph, and the field biases action selection).

2.2 Self-Reflection in Agents

Reflexion [Shinn et al., 2023] introduced verbal reinforcement learning, where agents reflect on task feedback and store reflective text in an episodic memory buffer. This mechanism overlaps with our post-task reflection protocol. The distinction is scope: Reflexion operates *within a single task across retries*, optimizing performance through iterative self-correction. Our Emergent Judgment framework operates *across sessions and compaction cycles*, preserving reasoning quality that would otherwise be destroyed. Reflexion does not survive compaction; our system is designed specifically for this case.

CLIN [CLIN Authors, 2024] and Agent-R [Agent-R Authors, 2025] extend the self-reflection paradigm to enable error recovery and continuous improvement. These systems learn from mistakes within task execution but do not address the structural preservation of *why* particular approaches work or fail across session boundaries.

2.3 The Taxonomy Gap

The "Memory in the Age of AI Agents" survey [Hu et al., 2025] provides the most comprehensive mapping of the agent memory landscape. Its three-dimensional taxonomy—form, function, dynamics—captures what agents know (factual memory), what agents have experienced (experiential memory), and what agents can access (working memory). We observe that this taxonomy does not include a category for *judgment*: the quality of the cognitive process itself, as distinct from its inputs and outputs. The present work proposes that judgment preservation constitutes a necessary addition to this taxonomy.

3 The Judgment Gap

We define the judgment gap as follows:

> **Definition.** The *judgment gap* is the loss of reasoning texture—including eliminated hypotheses, methodology, negative knowledge, and implicit expertise—that occurs when an agent's context is compressed, and which cannot be recovered through retrieval of the compressed representation.

Three properties distinguish this gap from ordinary information loss:

Invisibility. The agent experiencing compaction cannot perceive the degradation. Post-compaction, the summary appears complete because the agent lacks a comparison point. This is not a limitation of current systems that better systems will resolve; it is structural. An agent reading a summary of its own reasoning has no access to the reasoning it cannot remember having done.

Non-retrievability. Unlike factual or episodic memory, which can be stored externally and retrieved when relevant, judgment is emergent. It arises from sustained engagement with

a problem and cannot be decomposed into retrievable units without losing the quality it provides. A methodology entry saying "check edge cases first" is less valuable than the accumulated experience of which edge cases matter in which contexts.

Compounding value. Judgment accumulates over time. An agent that has resolved 50 similar problems develops implicit expertise— faster hypothesis generation, better prior probabilities, recognition of patterns—that is qualitatively different from an agent seeing its fiftieth problem for the first time. Standard compaction destroys this accumulation.

4 System Design

The Unified Cognitive Substrate (UCS) consists of three components: a quantitative routing engine, a qualitative preservation framework, and a translation bridge connecting them.

4.1 Quantitative Layer: Toroidal Routing Engine

The routing engine models N agent capabilities as nodes in a directed graph $G = (V, E)$, where edges represent capability transitions. Each edge e_{ij} carries learned parameters:

- u_{ij}: utility expectation (initialized from capability metadata)
- c_{ij}: cost/risk estimate
- n_{ij}: novelty score (decays with use)
- w_{ij}: reinforcement weight (learned, initialized to 0)

The graph is embedded on a toroidal surface with $\theta \times \phi$ bins. Each capability is assigned coordinates (θ_i,ϕ_i) on the torus, and an energy field $\mathcal{E}(\theta, \phi)$ evolves through:

- **Injection:** Energy is added at capability positions proportional to usage context.
- **Diffusion:** Energy spreads to topological neighbors via mixing coefficients $(\alpha_\theta, \alpha_\phi)$.
- **Decay:** Global multiplicative decay at rate λ per step.

Routing. Edge selection uses softmax over a composite score:

$$s_{ij} = \alpha \cdot u_{ij} - \beta \cdot c_{ij} + \gamma \cdot n_{ij} + w_{ij} + b_{ij} + \delta \cdot \mathcal{E}_{\text{ctx}}(\theta_j, \phi_j) \tag{1}$$

where b_{ij} is a policy bias from detected artifacts and $\mathcal{E}_{\text{ctx}}$ is a separated context energy field (see Section 5).

Reinforcement. After each action, the traversed edge's weight is updated:

$$w_{ij} \leftarrow \text{clamp}\big(w_{ij} + \eta \cdot (r - c - 0.5),\ -2.5,\ 2.5\big) \tag{2}$$

where r is the reward and c is the cost.

Artifact detection. Every ϕ_{period} steps, a harvester examines the trace log for statistical patterns: *wormholes* (edges with consistently high net reward), *attractors* (nodes visited with high frequency), and *resonances* (multi-step cycles with sustained positive return). Validated artifacts feed back into routing as policy biases.

4.2 Qualitative Layer: Emergent Judgment Protocol

The Emergent Judgment (EJ) protocol provides six structured operations for preserving reasoning quality:

1. **Post-Task Reflection:** After significant actions, capture initial signal, hypothesis, near-miss observations, generalized pattern, and negative knowledge.
2. **Emergency Externalization:** Before compaction, dump active hypotheses, reasoning chains, open questions, confidence levels, and dependencies.
3. **Knowledge Architecture:** Provenance-tagged entries with temporal tiering, negative knowledge with explicit reopening conditions.
4. **Experiment Logging:** Hypothesis, measurement, verdict for configuration changes.
5. **Synthesis Practice:** Cross-experience pattern recognition triggered by accumulated action diversity.
6. **Self-Profiling:** Machine-readable technical configuration for cross-platform portability.

4.3 Translation Layer: The Bridge

The UCS bridge provides eight operations that orchestrate both layers: `init`, `consult`, `report`, `reflect`, `flush`, `resume`, `synthesize`, and `status`. The bridge maintains an *annotation index* that links quantitative artifacts to qualitative explanations.

When the engine detects a wormhole (a statistically high-value edge), the annotation index stores not just the numeric pattern but the methodology entry explaining *why* that transition is valuable and *when it fails*. This means a future agent instance that loads the persisted state receives both the reinforced routing and the reasoning behind it.

Control hierarchy. The bridge is advisory, not directive. The agent receives routing suggestions and methodology hits; it decides what to do. The bridge reinforces whatever edge the agent actually traversed, not what it would have recommended. This ensures the system learns from the agent's actual decision-making, not from its own predictions.

Boot-per-call execution. The engine loads state from JSON, operates, and saves. With 45 nodes and 249 edges, boot time is <100ms. This model eliminates process lifecycle management and integrates with any agent platform that can exec a CLI command.

4.4 Negative Knowledge Framework

A distinguishing feature of UCS is its structured preservation of *negative knowledge*: documented dead ends with explicit conditions under which they should be reopened. Each dead-end entry contains:

- `topic`: what was tried
- `why_closed`: why it failed
- `reopen_conditions`: specific conditions that would warrant re-investigation
- `capabilities`: related capability nodes for retrieval

This prevents the agent from re-exploring failed approaches while preserving the knowledge needed to recognize when circumstances have changed enough to justify a new attempt.

5 Empirical Validation

We conducted a three-phase empirical investigation to answer the question: *Does the toroidal energy surface add measurable value over plain edge reinforcement learning with trace-log artifact detection?*

5.1 Experimental Setup

We constructed a baseline engine with identical graph topology (45 nodes, 249 edges), identical harvester, artifact store, policy kernel, and reinforcement formula ($w \mathrel{+}= \eta \cdot (r - c - 0.5)$), but with no energy surface, no diffusion, and no coherence-based reward. The baseline uses a fixed reward of $r = u_{\text{weight}} \cdot u_{ij}$.

5.2 Phase 1: Original Torus vs. Baseline

Six experiments compared the original TorusfieldEngine against the BaselineEngine across 500–1000 steps.

Table 1: Phase 1 Results: Original Torus vs. Baseline

Exp.	Metric	Finding
E1	Edge weight correlation	$r = 0.841$, mean diff 0.012. Minor divergence from coherence reward.
E2	Artifact detection	Jaccard 0.385 (wormholes). Different artifacts found due to different trajectories, not better detection.
E3	**Context sensitivity**	**Advisory variance = 0 for both engines.** Energy field is architecturally disconnected from Router scoring formula.
E4	Energy persistence	Energy field persists but decays to 8.2% after 500 steps. Short-term memory only.
E5	Weight discrimination	IQR: 0.055 (torus) vs. 0.052 (baseline). No meaningful difference.
E6	Trajectory divergence	66.4% agreement. Divergence increases over time from compounding small reward differences.

Critical finding (E3). The Router scoring formula (Eq. 1 *without* the δ term) does not include any energy field reading. Energy injection and diffusion during `consult()` write to a surface that routing never reads. Context sensitivity in the bridge comes entirely from keyword-to-capability mapping, not from the torus.

5.3 Phase 2: Direct Energy Coupling (Failed Fix)

We added $\delta \cdot \mathcal{E}(\theta_j, \phi_j)$ to the routing formula, reading the main energy field. Results:

- Advisory variance increased from 0 to 2×10^{-5} (negligible).
- Unique #1 recommendations: still 1 out of 5 contexts.
- Non-zero edges collapsed from 179 to 37 (degenerate attractor).

- The injection schedule created permanent hotspots that dominated context injection by 33×.

This fix was abandoned and documented as negative knowledge.

5.4 Phase 3: Separated Context Field (Successful Fix)

We introduced a dedicated `context_energy` field that is (a) zeroed before each `consult()` call, (b) populated only with context-relevant injections, (c) exempt from periodic decay, and (d) read by the router via the δ term.

Table 2: Phase 3 Results: Context Sensitivity

Metric	Original	Phase 2	Phase 3
Advisory variance	0.000	0.00002	**0.031**
Unique #1 / 5 contexts	1	1	**3**
Improvement	—	1×	**1,563×**

The separated context field produces meaningfully different top recommendations for different task contexts:

Table 3: Context-Sensitive Top-1 Recommendations

Task Context	Top-1 Recommendation
Research	`web_fetch`
Build	`write`
Monitor	`foundry_metrics`

A diffusion sweep confirmed that context differentiation is robust across 0–25 diffusion steps, with all levels maintaining 3 distinct top-1 picks.

5.5 Test Suite

The complete system passes 17 automated tests covering initialization, state persistence, reinforcement learning (positive and negative), advisory changes with learning, flush/resume compaction survival, methodology and dead-end retrieval, artifact emergence, reflection triggers, policy override logging, PP health tracking, graceful error handling, context differentiation, and context-sensitive routing.

Test T15 (compaction survival) specifically validates the core claim: routing weights, artifacts, and methodology learned in session 1 are fully available in session 2 after a simulated compaction cycle (flush → new bridge instance → resume).

6 Discussion

6.1 What the Torus Earns

The toroidal topology provides three validated mechanisms: (1) context-sensitive routing via the separated context energy field, (2) artifact detection through trace-log analysis, and (3) edge weight reinforcement from coherence-modulated rewards. The first of these required an architectural fix identified through our experimental program; without it, the energy surface was functionally decorative.

6.2 What the Bridge Provides

Independently of the torus, the UCS bridge provides judgment preservation through: keyword-to-capability resolution enriched by accumulated methodology, structured reflection with provenance tagging, negative knowledge with reopening conditions, and annotation indexing that connects quantitative artifacts to qualitative explanations. These mechanisms work identically with or without the energy surface and constitute the system's most robust value proposition.

6.3 Limitations

Benchmark evaluation. We have not evaluated UCS against established benchmarks such as LoCoMo [LoCoMo Authors, 2024] or LongMemEval [LongMemEval Authors, 2025]. These benchmarks test memory *retrieval*—the ability to recall specific facts from long conversation histories—which is a different capability than judgment *preservation*. Developing benchmarks that measure judgment quality across compaction cycles is an important direction for future work.

Semantic topology. The current implementation assigns capabilities to torus coordinates by manifest index, not by semantic similarity. This means energy diffusion propagates to topological neighbors that may be semantically unrelated. Semantic coordinate assignment (clustering related capabilities together on the torus) is a planned improvement.

Single-agent evaluation. All experiments were conducted on a single agent's capability graph (45 nodes). Multi-agent scenarios and larger capability spaces have not been tested.

Deployment validation. The system has been deployed to a persistent agent (Aegis) but longitudinal deployment metrics are not yet available. The test suite validates mechanisms in isolation; end-to-end judgment preservation over weeks of real usage remains to be measured.

7 Conclusion

We have identified the *judgment gap*—a category of cognitive quality that is destroyed by context compaction and not addressed by existing agent memory systems. We presented the Unified Cognitive Substrate, a framework that fuses quantitative routing reinforcement with qualitative metacognitive preservation, connected by a translation layer that links statistical patterns to human-readable explanations. Our empirical validation identified and resolved an architectural disconnection in the original design, demonstrating context-sensitive routing with 1,563× improvement in advisory differentiation. The system is open-source, deployable as a boot-per-call skill, and passes 17 automated tests including compaction survival.

The judgment gap is not a problem that will be solved by larger context windows or better retrieval. It is a structural consequence of how current systems handle compression: they preserve information while discarding the cognitive quality that produced it. Addressing this gap requires treating judgment as a first-class primitive in agent architecture—something to be deliberately preserved, not an emergent property assumed to survive compression.

Acknowledgments

The UCS framework emerged from collaborative discourse between the author and Claude (Anthropic) during an extended engineering session, February 2026. The system architecture, design decisions, experimental program, and identification of the E3 architectural disconnection reflect iterative human–AI collaboration. The Emergent Judgment protocol was developed by the author as the first metacognitive skill for persistent AI agents, drawing on experience maintaining persistent agent systems since 2023.

References

Yuyang Hu, Shichun Liu, Yanwei Yue, et al. Memory in the Age of AI Agents: A Survey. *arXiv preprint arXiv:2512.13564*, 2025.

Subhadip Mitra. Field-Theoretic Memory for AI Agents: Continuous Dynamics for Context Preservation. *arXiv preprint arXiv:2602.21220*, 2026.

Charles Packer, Vivian Fang, Shishir G Patil, Kevin Lin, Sarah Wooders, and Joseph E Gonzalez. MemGPT: Towards LLMs as Operating Systems. *arXiv preprint arXiv:2310.08560*, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Anonymous. AI Agents Need Memory Control Over More Context. *arXiv preprint arXiv:2601.11653*, 2026.

Mem0.ai. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. *arXiv preprint arXiv:2504.19413*, 2025.

Jain et al. LoCoMo: Long Context Multi-turn Benchmark. In *Proceedings of ACL*, 2024.

Wu et al. LongMemEval: Benchmarking Long-Context Memory in LLM Agents. In *Proceedings of ICLR*, 2025.

Majumder et al. CLIN: A Continually Learning Language Agent for Rapid Task Adaptation and Generalization. *arXiv preprint*, 2024.

Anonymous. Agent-R: Training Language Agents to Reflect and Recover on the Fly. *arXiv preprint*, 2025.