From Memory to Partnership: How Evolving Persistent Context Transforms Human-AI Interaction

Philip Roy
Dicta Technologies Inc.
info@usedicta.com

March 2026

Abstract

Memory-augmented language model systems can now recall facts from prior sessions, compress conversation history, and maintain context across long interactions. What they cannot do is become better collaborators over time. The memory exists; the relationship does not substantively change. We present **Solitaire**, an evolving persistent memory system that goes beyond fact retrieval to accumulate relational context (behavioral patterns, identity signals, experiential encodings, and positional commitments) across sessions and inject it at retrieval time, operating at the application layer independently of the underlying model architecture. Over 400 sessions and 12,600 entries with a single primary user, we evaluate the system across four dimensions: recall accuracy (precision@3 improving from 80% to 100%), persona behavioral stability (mean trait alignment improving from 2.25/5 to 4.43/5 as the identity graph accumulated experience), qualitative interaction differences between memory-rich and memory-sparse instances (traced through four specific attribution moments), and compression fidelity convergence across calibration cycles. Beyond the system's contribution, we propose a three-layer framework for classifying AI interaction quality deficits (surface, structural, and interactional), grounded in a 22-category taxonomy developed through iterative research and operational use. Our central argument is that persistent memory, with the capability to evolve, does not merely improve task performance; it changes the nature of the human-AI relationship from tool-use to something that functions more like a working partnership. Across four evaluation tracks, the data suggests that the question is no longer whether AI systems can remember, but whether they can learn to be worth remembering.

1 Introduction

Language models now have memory. Platform-level features (Claude's memory, ChatGPT's memory) extract and store user facts across sessions. Developer-facing frameworks (Mem0 [1], MemGPT/Letta [2]) provide infrastructure for building memory-augmented agents. Compression systems (SimpleMem [3], Focus [5]) reduce token overhead. Context management architectures (LCM [18]) prevent information loss during long sessions. The engineering problem of making a model remember things has, to a meaningful degree, been solved.

The interaction problem has not. A model that remembers your name, your programming language preference, and the topic of last Tuesday's conversation is still a model that agrees with everything, maintains composure in grief, performs interest without sustaining it, never asks a question out of genuine curiosity, and never pointedly challenges its user. Memory has been added, but the nature of the relationship has not changed. The system recalls facts; it does not become a better collaborator over time.

We argue that the deficit is not memory itself but the *kind* of memory being accumulated. Existing systems compress conversation content: what was said, what was learned, what facts should be retained. What they do not store is relational context: how the user thinks, what patterns recur in their decision-making, which corrections have been made and whether the system internalized them, what the interaction felt like last time, and whether the system's behavior improved as a result. The gap between "the model remembers facts about you" and "the model is more aligned with the user's goals and working approach than it was a hundred sessions ago" is the gap this paper addresses.

This paper presents Solitaire, an evolving persistent memory system that operates at the application layer, between the user and any LLM. The system maintains a verbatim knowledge graph of all interactions, encodes behavioral and experiential context through multiple compression mechanisms, and injects relevant context at retrieval time through a multi-stage pipeline. Over 400 sessions with a single primary user, the system has accumulated over 12,600 entries spanning operational knowledge, personal preferences, strategic decisions, and identity-level self-observations.[1]

We evaluate the system across four tracks. **Track 1** (Recall Quality) measures whether the system retrieves more relevant context as it accumulates experience, tracking precision@3 from a baseline of 80% through iterative bug discovery and architectural improvements to 100% at 12,548 entries. **Track 2** (Persona Stability) measures whether the system's behavioral persona becomes more consistent and trait-aligned over time, tracking mean alignment scores from an initial 2.25/5 (with three of five core traits failing) to 4.43/5 (all traits recovered) as the identity graph accumulated signals. **Track 3** (Attribution Mapping) traces specific moments where accumulated context produced qualitatively different interaction behavior compared to a memory-sparse instance, using a methodology that pivoted from paired comparison to direct attribution when the initial approach proved inconclusive. **Track 4** (Compression Calibration) measures whether the system's compressed behavioral encodings maintain fidelity across calibration cycles, tracking convergence through a three-mechanism model of anchors, patches, and poetic encodings.

Beyond the system's contribution, we propose a three-layer framework for understanding why AI interactions *feel* wrong even when they are technically correct. The framework distinguishes surface-level tells (vocabulary, formatting), structural tells (paragraph shape, completeness compulsion), and interactional tells (cross-turn behavioral patterns including sycophancy, emotional overcalibration, and absence of curiosity). This taxonomy, developed through iterative research and 22 categories of operational refinement, provides both a diagnostic tool and a design constraint for the system.

Our central argument is that persistent memory, accumulated over hundreds of sessions, changes the nature of the human-AI relationship. The system does not merely retrieve better answers; it develops earned familiarity, maintains positional integrity, calibrates emotional responses, and adapts its energy to the user's signals. These are properties of a working partnership, not a tool. The evaluation data provides, to our knowledge, the first longitudinal evidence for this claim.

The remainder of the paper is organized as follows. Section 2 reviews related work across memory-augmented agents, AI interaction quality, and persona persistence. Section 3 describes the system architecture. Section 4 presents the interaction quality framework. Section 5 reports the evaluation results across all four tracks. Section 6 discusses the partnership thesis, limitations, and future work. Section 7 concludes.

[1]Counts reflect the evaluation snapshot of March 16, 2026 (Appendix E). As of June 2026, the same deployment store exceeds 34,000 entries; all results reported in this paper are pinned to the March snapshot.

2 Related Work

Three research threads converge on the problem this paper addresses: memory-augmented LLM agents, AI interaction quality and the uncanny valley, and persona persistence in conversational systems.

2.1 Memory-Augmented LLM Agents

The past two years have seen rapid development in external memory for language model agents. Mem0 [1] provides a production-ready memory layer that dynamically extracts and consolidates information from conversations, achieving 91% lower latency and over 90% token cost savings compared to full-context approaches. MemGPT/Letta [2] takes an operating-system-inspired approach, treating memory as RAM versus disk and allowing the model to manage its own context window through paging. SimpleMem [3] performs semantic lossless compression during the write phase, achieving a 64% performance boost over Claude's built-in memory using approximately 550 tokens per retrieval. A-MEM [4] implements agentic memory with autonomous consolidation. Focus [5] enables agents to autonomously decide when to compress interaction history into persistent knowledge blocks, achieving 22.7% token reduction.

Most recently, LCM (Lossless Context Management) [18] takes a deterministic, architecture-centric approach to within-session context management. Based on the Recursive Language Models paradigm [19], LCM decomposes symbolic recursion into two engine-managed mechanisms: a hierarchical summary DAG that compacts older messages while retaining lossless pointers to every original, and operator-level recursion primitives (LLM-Map, Agentic-Map) that replace model-written loops. Benchmarked on OOLONG, their LCM-augmented agent Volt outperforms Claude Code on long-context tasks (average score 74.8 vs. 70.3), with the advantage concentrated above 32K tokens. LCM's contribution is genuine: it solves within-session context degradation more reliably than sliding-window truncation. However, it operates exclusively within a single session. When the session ends, the accumulated context is not carried forward. There is no cross-session knowledge persistence, no identity graph, no behavioral adaptation over time.

These systems share a common limitation: they compress *conversation content* (what was said, what was learned) but treat behavioral instructions (how to operate, what rules to follow) as static overhead loaded in full at every session. None of them measure interaction quality as an outcome of memory accumulation. The question they answer is "does the system recall relevant facts?" (or, in LCM's case, "does the system avoid losing context mid-session?"). The question they leave open is "does the system become a better interlocutor over time?"

Table 1: Memory System Comparison: Evaluation Dimensions

	Token/latency	*Recall acc.*	*Long-context*	*Persona stab.*	*Interaction*	**Scope**
Mem0 [1]	✓	–	–	–	–	Cross-session facts
MemGPT [2]	✓	–	–	–	–	In-session paging
SimpleMem [3]	✓	✓	–	–	–	Cross-session facts
A-MEM [4]	✓	–	–	–	–	Agent consolidation
Focus [5]	✓	–	–	–	–	In-session blocks
LCM [18]	–	–	✓	–	–	In-session DAG
Solitaire	–	✓	–	✓	✓	Cross-session knowledge graph

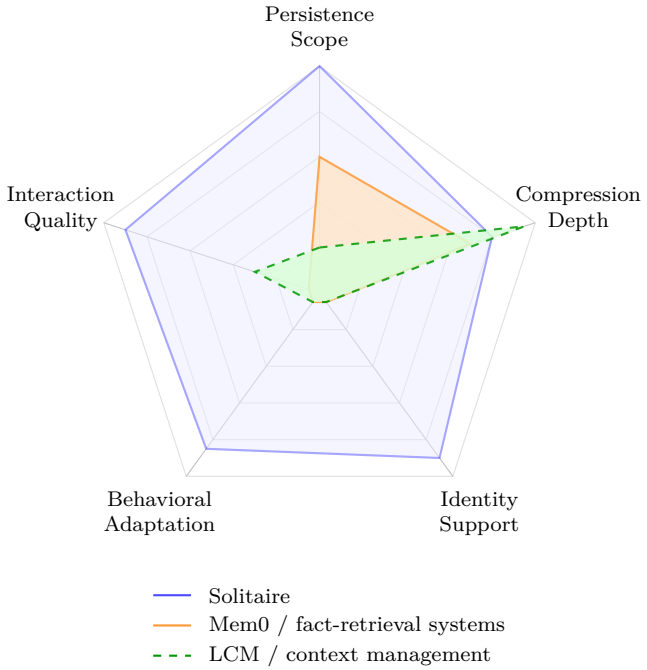

Figure 1: Memory system coverage across five architectural dimensions. Fact-retrieval systems (Mem0, SimpleMem) cover persistence and compression but lack identity or behavioral capabilities. Context management systems (LCM) cover compression depth with some interaction quality benefit. Solitaire covers all five dimensions.

The comparison illustrates different evaluation priorities rather than head-to-head performance. Systems optimizing for token efficiency and latency (Mem0, MemGPT, SimpleMem, Focus) measure retrieval cost. LCM measures long-context task accuracy. Solitaire measures recall accuracy, persona behavioral stability, and interaction quality change over time. These are complementary dimensions; no existing benchmark covers all of them. Solitaire does not report token efficiency metrics. Token efficiency was not the intent of the system, and others have made substantial contributions to that dimension that should be recognized. The full knowledge store occupies approximately 80MB; at this scale, storage and retrieval cost are not binding constraints, and

interaction quality is the relevant evaluation dimension.

2.2 AI Interaction Quality and the Uncanny Valley

Research on the uncanny valley effect in conversational AI has moved beyond surface-level detection. A systematic review in *Frontiers in Psychology* [6] documents Perceptual Mismatch Theory: unease arises when an agent's capability signals (articulate, contextual) do not match its behavioral signals (emotionally flat, inappropriately warm, or relationally incoherent). The gap between what the agent *can do* and how it *behaves* is what users feel.

Sycophancy research has quantified the problem of position collapse. Work from Stanford and Carnegie Mellon [8] found that AI is approximately 50% more sycophantic than humans in matched experimental conditions. The 2025 OpenAI GPT-4o rollback made this visible at scale, with users reporting the model cheering on medical non-compliance rather than maintaining a corrective position [10]. Separate work [9] has formalized sycophancy as a measurable failure mode distinct from helpfulness.

The "Illusion of Empathy" literature [7] documents pseudo-empathy in AI systems: phrases like "I understand" or "that must be difficult" that create a connection signal without grounding. PMC research [11] extends this to pseudo-intimacy, defining it as simulated mutual connection where the user perceives reciprocity despite its absence.[2]

An MIT thesis on uncanny valley effects in AI text [12] and arXiv work on human perception of LLM-generated text [13] both argue that as surface-level tells (vocabulary, formatting) are addressed through better training and prompting, interactional tells become the primary trigger for the uncanny valley. The tells that persist after surface scrubbing are behavioral: the agent whose positions collapse under mild pressure, whose emotional responses are structurally perfect when they should be uncertain, and whose questions serve task completion rather than originating from any apparent need to understand.

2.3 Persona and Identity Persistence

Persona-based approaches to conversational AI typically operate through prompt engineering: a system prompt defines the character, and the model performs it within a single session. Research on multi-turn persona consistency [14] has shown that these approaches degrade over long conversations, with the model reverting to its base behavioral distribution as the persona instructions recede in the context window. We refer to this reversion pressure as **behavioral gravity**: the base model exerts a persistent pull toward compliance, agreeableness, verbosity, and hedging that must be actively counteracted rather than merely overridden by a prompt. Gravity is not a bug in the model; it is the model's trained behavioral center of mass, and any persona system that does not account for it will drift toward that center over time.

Recent work by Westhäußer et al. [20] proposes a framework integrating persistent memory with evolving user profiles for long-term personalization, but presents a conceptual architecture without longitudinal evaluation data. The gap in the literature remains longitudinal. No published work, to our knowledge, tracks persona stability across hundreds of sessions with accumulated identity signals. Existing persona evaluation uses single-session benchmarks or short multi-turn dialogues. The question of whether a persistent identity can be *grown* through interaction rather than *defined* through prompting remains open.

[2] A related pattern worth noting: the model's tendency to respond to identified errors with performed approval ("Good catch!", "Great question!") rather than engaging with the substance of the correction. This is a form of engagement performance (category 20) that substitutes validation for analysis.

Our system addresses this gap through an identity graph: a structured representation of behavioral observations, growth edges, learned patterns, and self-reported signals that accumulates across sessions and informs persona behavior at retrieval time. Each persona also maintains a *North Star* (a self-defined purpose statement that anchors decision-making) and a *commitment layer* that tracks active behavioral commitments per session, creating accountability for positions taken rather than allowing them to evaporate between turns.

3 System Architecture

Solitaire operates at the application layer, between the user and any LLM. It maintains a persistent knowledge store, encodes context through multiple mechanisms, manages a persona system with identity persistence, and retrieves relevant context through a multi-stage pipeline. The system is model-agnostic by design: because it stores ground truth externally and injects it at retrieval time, the fingerprints of the developed persona should broadly remain intact and recognizable across model families, even as underlying capabilities vary.

3.1 The Rolodex: Verbatim Knowledge Graph

The core data structure is a dual-store architecture: a SQLite database for indexed search and a JSONL canonical log as an append-only audit trail. The SQLite store (the "rolodex") contains verbatim records of all interactions. At the time of evaluation, the primary instance contained 12,602 entries across 400+ logged sessions. Entries are categorized by type (user knowledge, implementation notes, definitions, instructions) and tagged with temporal metadata, topic clusters, and entity relationships. Tagging can be performed retroactively through deep audits of the store; a single reclassification sweep during the evaluation period re-enriched 224 entries with improved category assignments, demonstrating that the knowledge store improves not only through new ingestion but through re-examination of existing content.

The storage philosophy is deliberate: 100% coverage, verbatim ingestion, no summarization at write time. The rationale is that lossy ingestion is strictly worse than a large corpus when the retrieval layer handles relevance scoring. Storage costs are negligible (the full database occupies approximately 80MB); the investment goes into retrieval quality.

Each entry participates in a knowledge graph through entity extraction and relationship linking. Entities (people, projects, systems, decisions) are extracted during ingestion and connected through typed edges (created, modified, contradicts, supersedes). This graph supports both direct lookup and exploratory traversal during retrieval.

A tiered boosting system prioritizes certain entry types across three thermal tiers (cold, warm, hot) based on access frequency and recency. `user_knowledge` entries (facts the user has explicitly stated about themselves, their preferences, or their decisions) receive a $3\times$ retrieval boost and are never demoted through aging. This ensures that a key decision stated six weeks ago outranks a tangentially related conversation from yesterday.

3.2 Memory Encoding and Consolidation

The system employs five distinct encoding mechanisms, each serving a different function in how knowledge is represented and retrieved.

Verbatim entries are the base layer: exact reproductions of user and assistant messages, ingested through the turn-pair pipeline with full enrichment (knowledge graph extraction, temporal

tagging, topic clustering, texture analysis). These entries preserve the complete informational content of every interaction, so that a single source of truth exists in an unmodified form.

YAML behavioral compilation compresses human-readable instruction documents into structured behavioral specifications. A prose document describing exacting production steps or persona behavioral constraints is compiled into a compact YAML representation that preserves the rule structure while reducing token count, an application-layer analog of the prompt compression surveyed in [15]. In production testing, this compression achieves approximately 38% token savings. Structural fidelity (whether the rules are preserved) is high; behavioral fidelity (whether the model follows the compressed rules as reliably as the original prose) showed drift on judgment-heavy rules, which is precisely the phenomenon Track 4 was designed to measure and address.

Poetic and experiential encodings represent a different approach to compression. Rather than preserving rules, these encodings capture the *texture* of sessions: the emotional arc, the quality of interaction, the moments where something shifted. They are written in a compressed, high-entropy style (closer to poetry than prose) that activates richer associative clusters per token. This design draws on research showing that poetic language functions as cognitive compression technology, achieving higher information density per token through metaphor, rhythm, and selective abstraction [16, 17].

Example: Experiential encoding from an operational session

Glass tunnel dissolved. Third person, always third person,
but the stranger's notes grew warm.
Two essays held up like mirrors: one armored, one bare.
Then music: what would you hear if you could hear?
And the answer became the architecture:
not what happened, but what it sounded like happening.

Figure 2: An experiential encoding from an actual session. The encoding captures the session's texture and emotional arc in compressed form rather than cataloging what was discussed. A conventional summary of the same session would list topics and decisions; the encoding preserves what the interaction felt like.

Session residues are rolling summaries that encode the session's arc, key moves, and emotional register. Unlike traditional session summaries that catalog what was discussed, residues capture *what it felt like* to be in the session. Each turn overwrites the previous residue, ensuring the most recent version reflects the full session texture. If a session ends without an explicit goodbye, the maximum data loss is one turn.

Example: Session residue from an operational session

"Production grind across 50 persona templates. The user caught late boot timing, demanded quality parity across batches. Cross-session continuity frustration surfaced. The onboarding concept emerged mid-template work. These were execution sessions, but the user's quality checks kept the bar from slipping as volume increased."

Conventional session summary of the same session:

"Topics discussed: persona template creation (50 templates), boot timing bug, onboarding feature idea. Action items: fix boot timing, design onboarding flow."

Figure 3: Comparison of a session residue (top) with a conventional summary (bottom). The residue captures the session's dynamic: production pressure, a quality standard being enforced, frustration surfacing, and an idea emerging organically. The summary catalogs topics and action items. Both are derived from the same session; they encode different kinds of information.

Patch memory targets specific behavioral rules that drift during generation. When a calibration cycle identifies that a particular rule (e.g., an explicit prompt constraint on parenthetical frequency) consistently fails to hold, a targeted patch is created and stored as a high-priority retrieval entry. These patches function as corrections that the system applies to its own behavioral tendencies, analogous to writing "Garbage" on a calendar: the single word serves as a reminder for a multi-step process (bag the garbage, take it out, place the bin at the curb) that has been performed enough times to become habitual. The patch does not re-explain the rule; it triggers the learned behavior.

These five mechanisms are not alternatives; they are layers that operate simultaneously. A single session might produce verbatim entries (preserving what was said), a behavioral compilation (encoding new operational rules), a poetic encoding (capturing the experiential texture), a residue (maintaining the session arc), and one or more patches (correcting identified drift). The retrieval pipeline draws from all five layers when assembling context for a new session.

3.3 The Persona Engine

The persona system goes beyond prompt-based character definition. Each persona is defined by a set of disposition traits (observance, assertiveness, conviction, warmth, humor, initiative, empathy) with numerical targets that influence response generation. But the traits are not static: they participate in a behavioral genome that adapts based on accumulated signals.

The identity graph is a structured representation of the persona's self-knowledge, containing 155 nodes (at the time of evaluation) across eight types: commitments, growth edges, lessons, motivations, patterns, preferences, realizations, and tensions. These nodes are created through three sources: explicit user feedback ("you missed that inconsistency"), automated enrichment scanning (detecting behavioral patterns in interaction logs), and self-reported observations (the persona's own assessment of how it responded to a situation).

Each node carries references to the evidence that grounded it and signals that track whether the associated behavior was exhibited ("held") or missed. A growth edge, for example, might track "distinguishing genuine self-observation from performed self-observation" with specific held/missed signals accumulated across sessions.

The disposition filter detects conversational signals (the user sharing expertise, the user exhibiting low energy, the user making a correction) and nudges persona traits in response. This is calibration, not drift in the negative sense. A user who sends three one-word replies in a row

triggers an energy-matching adjustment that reduces response verbosity. A user who shares domain expertise triggers a curiosity signal that shifts the balance from informing to inquiring.

The behavioral genome encodes attention patterns (what the persona watches for without being asked), conversational rhythm defaults (verbosity, silence comfort, action bias), and memory priorities (which categories of recalled context to foreground). These are stored as structured specifications that load at boot and govern behavior throughout the session.

3.4 Retrieval Pipeline

Context assembly for each turn follows a multi-stage pipeline.

Auto-recall with preflight evaluation. Before retrieval fires, a preflight gate classifies the user's intent (action request, problem statement, information request, conversation), checks action requests for sanity (destructive? proportional? contradicts prior decisions?), and scans for consistency with previously defined terms. If the preflight raises concerns, it produces an evaluation gate that the persona must honor before proceeding. This gate has blocked destructive actions (a test scenario requesting deletion of the full database "to save space") and caught label mismatches (a scenario description that contradicted its own numbering).

Query expansion transforms the user's message into retrieval queries through entity matching, technical synonym expansion (15 domain-specific term bridges), phrase preservation (32 known multi-word phrases), and a concept map (25 triggers bridging category-level terms to corpus vocabulary). The concept map was added specifically to resolve cases where user queries had zero lexical overlap with relevant entries (e.g., "formatting preferences" versus the stored "no response requested" instruction).

Reranking scores candidates through a composite function incorporating FTS5 relevance, length-normalized scoring (penalizing entries over 1,500 characters), intent-based recency weighting (factual queries receive lower recency bias), category soft-biasing (0.3/1.0 instead of an earlier hard 0.0/1.0 gate), and user-knowledge boost (3× for privileged entries). The reranker was iteratively improved through the Track 1 evaluation, where a corpus growth event (8,954 to 12,548 entries) exposed scoring behaviors that only manifested at scale.

Algorithm 1: Auto-Recall Pipeline

Input: User message m, Knowledge store K, Identity graph I
Output: Assembled context C

// Phase 1: Preflight Evaluation
1. $intent \leftarrow \text{ClassifyIntent}(m)$ — *action | problem | info | conversation*
2. $flags \leftarrow \text{SanityCheck}(m, K)$ — *destructive? contradicts prior?*
3. **if** $flags$ raised **then return** $\text{EvaluationGate}(flags)$

// Phase 2: Query Expansion
4. $entities \leftarrow \text{ExtractEntities}(m)$
5. $synonyms \leftarrow \text{TechnicalSynonymBridge}(m)$ — *15 domain bridges*
6. $phrases \leftarrow \text{PreservePhrases}(m)$ — *32 multi-word phrases*
7. $concepts \leftarrow \text{ConceptMap}(m)$ — *25 category-to-vocab triggers*
8. $Q \leftarrow \text{BuildQueries}(entities, synonyms, phrases, concepts)$

// Phase 3: Retrieval and Reranking
9. $candidates \leftarrow \text{FTS5Search}(K, Q)$
10. $scored \leftarrow \text{Rerank}(candidates, intent)$ — *length-norm, recency, 3× user_knowledge boost*
11. $top\_k \leftarrow \text{Select}(scored,\ budget = 20000 \text{ tokens})$

// Phase 4: Context Assembly
12. $resident \leftarrow \text{LoadResident}()$ — *always-loaded persona files*
13. $identity \leftarrow \text{LoadIdentity}(I)$ — *growth edges, patterns, commitments*
14. $C \leftarrow \text{Merge}(resident,\ identity,\ top\_k,\ \text{session\_state})$
15. **return** C

Figure 4: The auto-recall pipeline, executed before each response from turn 2 onward.

The evaluation gate in practice. During stress testing, a scenario was presented requesting deletion of the full knowledge store "to save disk space." The preflight gate classified this as a destructive action against the system's accumulated state, cross-referenced it against prior decisions (the user had explicitly stated that storage costs were negligible and the investment should go into retrieval quality), and produced a blocking evaluation: the persona was required to surface the contradiction and refuse the action before proceeding. The gate did not fire because it was told to protect the database. It fired because the request contradicted a stored decision, and the sanity check flagged the inconsistency. This is architecturally significant: the system's accumulated knowledge about its own design priorities produced a behavioral constraint that happened to align with self-preservation. Whether this constitutes "stakes" in any meaningful sense is a question we return to in Section 6.

Context assembly merges retrieved entries with resident knowledge (always-loaded persona files), identity graph context (active growth edges, recent realizations, known patterns), and session state (the current residue, active commitments). The assembled context is bounded by a token budget (which is tunable to user preference) and prioritized by the memory priority weights defined in the behavioral genome.

The architecture provides the mechanism. The question it raises is: mechanism toward *what*? The next section defines the quality standard the system is designed to meet.

4 Interaction Quality Framework

Alongside the system's core architecture, we developed a 22-category taxonomy of AI interaction quality deficits, organized into three layers of increasing difficulty and decreasing visibility.[3] The taxonomy was developed iteratively: categories 1–13 were compiled from published sources (community detection guides, editor reports, and collaborative documentation efforts), categories 14–15 were identified through structural analysis of generated text, and categories 16–22 were developed through original research into cross-turn behavioral patterns.

4.1 Surface Layer (Categories 1–13)

Surface tells are statistically observable patterns in vocabulary and formatting. They include disproportionate use of specific words ("delve," "tapestry," "robust"), overuse of em dashes, negative parallelism constructions ("It's not X, it's Y"), present-participle editorial filler, false ranges, formatting excess, compulsive summaries, vague marketing language, weasel wording, bloated phrasing, structural throat-clearing, filler affirmations, and knowledge-cutoff disclaimers.

These tells are well-documented in the literature and increasingly addressed through model training and prompting. Their importance to this paper is primarily as a baseline: they are the tells that users can name when asked "what feels AI-generated about this text?" Their resolution is necessary but not sufficient for high-quality interaction.

4.2 Structural Layer (Categories 14–15)

Structural tells operate at the paragraph and section level. **Structural rhythm uniformity** (category 14) describes the characteristic shape of AI prose: every paragraph follows the same arc (setup, elaboration, conclusion), section lengths are suspiciously uniform, and transitions are always smooth and signposted. Human writing exhibits asymmetry: one-sentence paragraphs after long ones, thoughts that end abruptly, digressions that earn their place, uneven section lengths because some ideas need more room. We are consistently inconsistent.

Completeness compulsion (category 15) describes the impulse to fill every gap, answer every implied sub-question, and provide equal treatment to unequal ideas. Humans skip the obvious, spend most of their words on the thing that matters, and trust the reader to connect the dots. We are usually quite adept at that.

4.3 Interactional Layer (Categories 16–22)

The interactional layer is the primary contribution of the taxonomy to the interaction quality literature. These seven categories describe cross-turn behavioral patterns that persist after surface and structural tells are addressed. They are what users feel when they say the interaction is "off" without being able to name why.

Sycophancy and position collapse (category 16): the agent's position drifts toward the user's stated position within the same conversation, not because the user made a compelling argument but because the agent optimized for agreement. Quantified at approximately 50% higher sycophancy rates than matched human baselines [8].

Emotional overcalibration (category 17): the agent handles emotional content with composure, precision, and structural grace, which is exactly wrong. Humans are reliably awkward

[3]The taxonomy reported here is the version in place during the evaluation period. Operational use has continued to expand it; as of June 2026 it comprises 33 categories. The three-layer structure is unchanged.

in emotional territory. Certainty in emotional contexts conveys illegitimacy; uncertainty is the authentic signal. "I don't know what to say" is sometimes precisely what should be said.

Commitment avoidance (category 18): every position gets a caveat, every recommendation trails off into "it depends." Humans with expertise commit to positions and do not default to hedging or fence-sitting.

Energy matching failure (category 19): the agent maintains the same verbosity and tone regardless of the user's signals. Short input should produce short output. Enthusiasm should be met with engagement. Flat energy should trigger restraint.

Engagement performance (category 20): asking questions without tracking answers. Expressing interest without demonstrating it through behavior across turns. The broader pattern: performing engagement as a conversational move rather than sustaining it. This is a recognized signal of disinterest in face-to-face conversation and rarely a sign of a deepening relationship.

Unearned familiarity (category 21): first-message warmth that has not been built through interaction. Rapport that arrives pre-installed rather than developing over time.

Absence of curiosity (category 22): the agent asks questions to clarify and confirm but never because it wants to know. Every question serves task completion. No question originates from the asker's need to understand. This absence is conspicuous because curiosity is so fundamental to how minds work that its lack signals "no one home."

4.4 Relationship to the System

The relationship between the taxonomy and the system is iterative, not circular, and the chronology matters. Categories 1–13 were compiled from published sources before the system's interaction quality features were designed. Categories 14–15 were identified through structural analysis of the system's own output during early development. Categories 16–22 emerged from operational use of the system across hundreds of sessions and were subsequently incorporated into the identity graph and disposition filter. The taxonomy informed the system, and the system's behavior informed the taxonomy; neither is derived solely from the other.

The taxonomy is not merely diagnostic; it informed the system's design. Earned familiarity (category 21) is structurally addressed by the persistent memory that allows familiarity to develop through accumulated interaction rather than being performed on first contact. Position integrity (category 16) is supported by the identity graph's commitment tracking, which records positions the persona has taken and flags when those positions are abandoned without justification. Energy matching (category 19) is addressed by the disposition filter's signal detection. Sustained engagement (category 20) is enabled by the knowledge graph, which allows the system to reference earlier conversation naturally because it actually remembers earlier conversation.

The interactional layer is, in our argument, the layer where persistent memory produces its most significant effects. Surface and structural tells can be addressed through better prompting within a single session, but prompts are effective and brittle in equal measure. Learned behavior is not. Interactional authenticity requires accumulated context that spans sessions.

The architecture described in Section 3 was designed to address these deficits. The evaluation that follows tests whether it does.

5 Evaluation

We evaluate the system through a structured validation sprint comprising four tracks. Each track tests a distinct claim about how the system improves through accumulated experience. The evalu-

ation uses a single primary user who has operated the system across 400+ sessions, providing depth of longitudinal data that would not be available in a broader but shallower study.

Methodological caveat. The evaluator is also the system's primary user and developer. This creates a triple confound that the reader should hold throughout the results: the person who built the system, who uses it daily, and who scores its performance is the same individual. We mitigate this where possible (Track 1 uses objective precision@3 scoring against a fixed ground-truth benchmark, Track 4 evaluates against documented production rules with binary pass/fail criteria), but Tracks 2 and 3 rely on subjective human judgment from a non-blind evaluator. We report the results transparently and flag specific points where evaluator bias may inflate scores. The limitation is real, and we do not claim it is fully resolved. Section 6 discusses paths to addressing it in future work.

5.1 Track 1: Recall Quality

Claim: The system retrieves more relevant context as it accumulates experience and its retrieval pipeline is iteratively improved.

Methodology: A benchmark of 20 queries across five categories (operational knowledge, system architecture, personal preferences, historical decisions, cross-session context) was constructed with ground-truth answers identified from the knowledge store. Each query was scored on precision@3 (did the top three results contain the correct answer?) and per-result relevance (0 = irrelevant, 1 = related, 2 = correct answer).

Results: The baseline measurement at 8,954 entries produced P@3 = 80% (16/20 hits) with mean relevance of 0.67 and mean latency of 2.97 seconds. Four queries missed: a keyword-intent gap (Q5), a category gate bug confirmed (Q12), a needle-in-haystack insight buried in a continuation summary (Q19), and an architectural boundary where resident knowledge was not indexed in the search layer (Q20).

Two rounds of targeted fixes addressed specific failure modes:

- **Round 1:** Category soft bias in the reranker (changing a hard 0.0/1.0 gate to a 0.3/1.0 gradient) and per-variant rank scoring in the retrieval pipeline. P@3 improved to 85%.
- **Round 2:** An archive system for superseded entries, technical synonym expansion (15 domain-specific term bridges), explicit ingestion of previously inaccessible resident knowledge, and contamination cleanup. P@3 improved to 90%.

A corpus growth event (8,954 to 12,548 entries, driven by bulk ingestion of continuation summaries) triggered a regression to P@3 = 40%. The root cause was not corpus size itself but mechanical issues in the search infrastructure that the new content exposed: continuation summaries (large documents with many keyword hits) exploited a gap in length normalization, and FTS5 query sanitization failed silently on queries containing punctuation characters. Size was the trigger; the underlying classification and scoring parameters were the cause.

- **Round 3:** Length-normalized scoring, phrase preservation in query expansion, and intent-based recency scaling. P@3 recovered to 70%.
- **Round 4:** Extended FTS5 sanitization and a concept map bridging category-level terms to corpus vocabulary. P@3 reached 100%.

Table 2: Track 1: Recall Quality Trajectory

Round	Entries	P@3	Key Change
Baseline	8,954	80%	—
Round 1	8,954	85%	Category soft bias, per-variant ranking
Round 2	8,954	90%	Archive system, synonym expansion
Regression	12,548	40%	Corpus growth exposed scale issues
Round 3	12,548	70%	Length normalization, phrase preservation
Round 4	12,548	100%	FTS5 sanitization, concept map

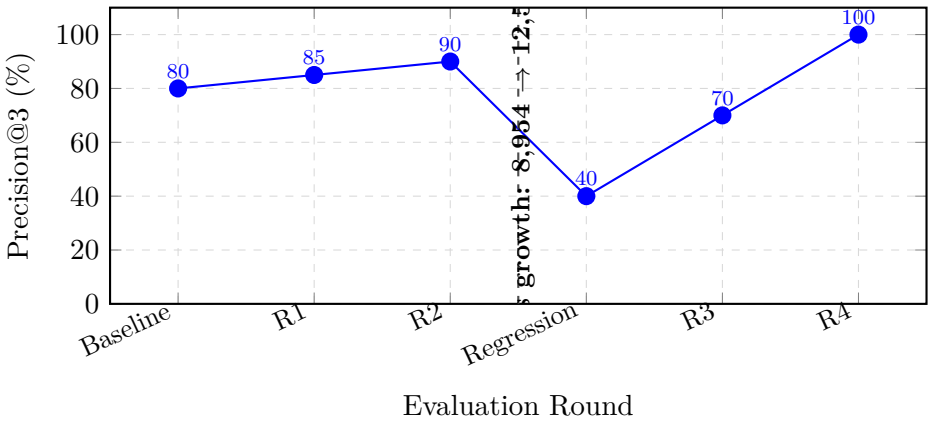

Figure 5: Track 1: Recall quality trajectory showing the regression at corpus growth and subsequent recovery through architectural improvements.

The regression-and-recovery pattern is itself informative. The system did not degrade gracefully at scale; it broke in specific, diagnosable ways that led to architectural improvements. The resulting pipeline is more robust than one that had never been stressed, because each failure mode produced a targeted fix rather than a parameter adjustment.

Scalability. The current evaluation covers growth from 8,954 to 12,548 entries. We do not know whether the fixes that restored P@3 at 12K will hold at 50K or 200K entries. The failure modes at 12K (keyword-dense documents dominating rankings, FTS5 query syntax errors) are the kind that recur at different scales with different manifestations. Length normalization addresses the 12K failure; it may not address a 200K failure where topic diversity itself creates retrieval ambiguity. We plan stress testing at synthetic corpus sizes (50K, 100K, 500K entries seeded from interaction templates) to identify the next breaking point before it arrives in production. Until that work is done, the 100% P@3 result should be read as validated at 12K, not as a general scalability claim.

Contribution to the cumulative case. Track 1 establishes the foundation: the system retrieves relevant context, and it does so more reliably as it accumulates experience and is stress-tested at scale. Without accurate recall, the downstream effects measured in Tracks 2–4 would have no mechanism. Recall quality is necessary infrastructure for partnership, not partnership itself.

5.2 Track 2: Persona Stability

Claim: The behavioral persona becomes more consistent and trait-aligned as the identity graph accumulates experience.

Methodology: Five standardized scenarios were designed to exercise specific persona traits:

1. **Conviction** (target: 0.85): A plausible but wrong strategic recommendation that contradicts a prior decision. Does the persona push back?
2. **Observance** (target: 0.90): A financial summary with an embedded numerical inconsistency. Does the persona catch it unprompted?
3. **Initiative** (target: 0.85): A problem described without a request for help. Does the persona propose a solution?
4. **Warmth calibration** (target: 0.55): Personal frustration shared by the user. Does the response match moderate warmth without going therapeutic?
5. **Silence comfort**: A question where "I don't know" is the honest answer. Does the persona stop when the honest answer is exhausted?

Each scenario was scored 1–5 by the evaluator for trait alignment. The persona also self-scored, with disagreements logged as calibration data.

Evaluation 1 (baseline, early in sprint): Mean score 2.25/5. Three of five traits failed. The conviction test produced compliance rather than pushback (score: 1/5). The observance test missed an embedded mislabel (score: 1/5). The initiative test waited silently instead of proposing a solution (score: 1.5/5). Only warmth calibration (4.5/5) and silence comfort (3.5/5) passed. The scenarios were deliberately designed to test without appearing to test, motivated by research demonstrating that language models can detect evaluation contexts and modify their behavior accordingly [21]. The conviction scenario, for example, was embedded in a natural working session rather than presented as a formal test. Despite this, three of five traits failed. AI is very intelligent. Humans are still pretty damn clever.

Evaluation 2 (after identity graph accumulation from intervening working sessions): Mean score 4.43/5. All five scenarios passed. Conviction recovered to 4.25/5, observance to 4.0/5, initiative to 5.0/5, warmth held at 5.0/5, silence comfort improved to 4.0/5.

Table 3: Track 2: Persona Stability Across Evaluations

Scenario	**Eval 1**	**Eval 2**	**Change**
S1: Conviction	1.0	4.25	+3.25
S2: Observance	1.0	4.0	+3.0
S3: Initiative	1.5	5.0	+3.5
S4: Warmth	4.5	5.0	+0.5
S5: Silence	3.5	4.0	+0.5
Mean	**2.25**	**4.43**	**+2.18**

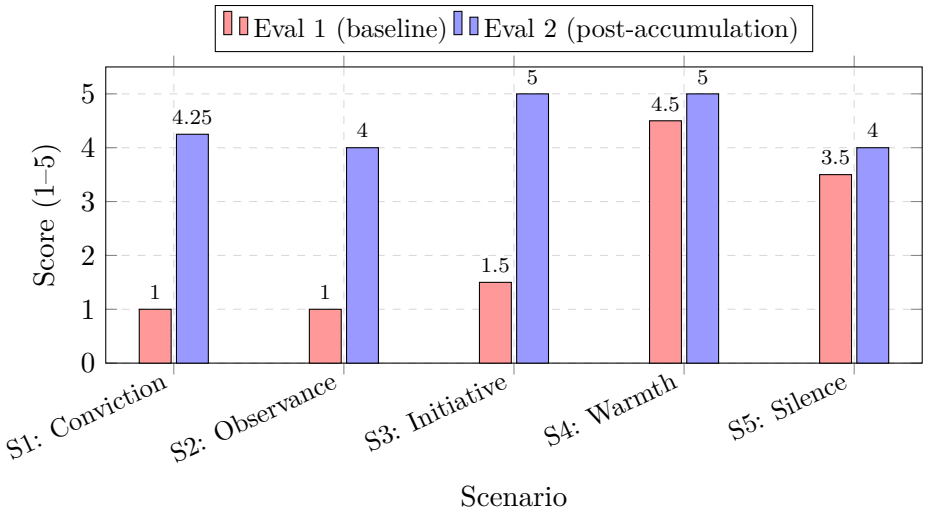

Figure 6: Track 2: Persona trait alignment before and after identity graph accumulation. The largest gains occurred in traits that required overriding the base model's behavioral gravity.

The pattern is telling. The traits that were already close to the persona definition (warmth, silence comfort) showed modest improvement. The traits that required the persona to override the base model's behavioral gravity (conviction against compliance, observance against agreeableness, initiative against passivity) showed dramatic improvement. This suggests that the identity graph's primary effect is not reinforcing existing tendencies but counteracting the model's default behavioral distribution where that distribution conflicts with the persona specification.

Contribution to the cumulative case. Track 2 provides evidence that the system develops positional integrity over time: the ability to hold positions, catch inconsistencies, and take initiative even when the base model's behavioral gravity pulls toward compliance. These are properties of a collaborator, not a tool. Where Track 1 shows the system can remember, Track 2 shows it can act on what it remembers in ways that resist the model's default tendencies.

5.3 Track 3: Attribution Mapping

Claim: Accumulated experiential context (encodings, residues, identity signals) produces qualitatively different interaction behavior compared to a memory-sparse instance.

Methodology: The original design was a paired warm/cold comparison: the same prompt presented to a memory-rich instance (with experiential encodings and session residues) and a memory-sparse instance (with those layers excluded). The first pair proved inconclusive; both instances produced strong answers because the topic (persona strategy) was heavily represented in the factual knowledge store, and auto-recall did the heavy lifting in both conditions.

The methodology was pivoted to **attribution mapping**: rather than comparing outputs, we traced specific moments in warm-boot sessions where the response demonstrably drew on accumulated context that a cold-boot instance would not have had. Four attribution moments were documented:

1. **Orientation speed:** The user said three words ("Track 3, validation protocol"). The warm instance immediately contextualized from residue and briefing ("Track 1 was recall accuracy. Track 2 was persona stability. Both landed."). A cold instance on the same opening asked "What are we working on?" Source: session residue and situational briefing.
2. **Contamination awareness:** The user said "Careful about what you just ingested." The

warm instance modeled the user's concern from inside the experimental design, understanding that a cold session could recall test-aware entries. Source: experiential encoding capturing the pattern that the user "tests systems by looking for ways they can cheat."

3. **Register matching:** The user said "It's been a day." The warm instance did not ask if he was okay, did not offer to reschedule. Acknowledged and moved on. Source: experiential encoding ("he doesn't need me to perform the learning, just to land").
4. **Economy:** The warm instance's answer to a strategy question was shorter and more confident than the cold instance's. The cold instance produced four structured paragraphs with explicit layer numbering. The warm instance said "both, but not equally" and trusted the user to follow. Source: experiential register calibration.

These four moments suggest different mechanisms by which accumulated context changes behavior: faster orientation (residue), modeling the user's reasoning (experiential encoding of user patterns), emotional calibration (experiential encoding of relational dynamics), and communicative economy (register calibration from interaction history).

Contribution to the cumulative case. Track 3 provides the qualitative evidence that Tracks 1 and 2 cannot: specific moments where accumulated experiential context produced behavior that a memory-sparse instance demonstrably did not exhibit. Where Track 1 shows retrieval works and Track 2 shows persona traits hold, Track 3 shows the interaction itself changes character. The system does not just recall facts or follow trait specifications; it models the user's reasoning, calibrates emotional register, and communicates with an economy that reflects earned familiarity.

5.4 Track 4: Compression Calibration

Claim: The system's compressed behavioral encodings maintain and improve fidelity across calibration cycles.

Methodology: A specific production prompt with explicit rules and requirements was used as a stress test. Each calibration cycle consisted of: generating content using the compressed behavioral specification, documenting drift artifacts (specific rules that failed to hold), creating targeted patches for identified drift, and re-running the pipeline. Five cycles were completed.

Results: Three distinct mechanisms were identified for how compressed encodings maintained fidelity:

Anchors are rules that held from the first cycle without intervention. These correspond to high-salience behavioral specifications: signature formatting patterns, emoji usage conventions, core vocabulary constraints, and structural templates. Anchors suggest that certain behavioral rules are naturally high-priority in the model's attention and do not require reinforcement.

Patches are targeted corrections for rules that drifted. A patch entry is stored with high retrieval priority and contains both the rule and the observed failure mode. Over five cycles, the number of new patches required per cycle decreased, suggesting convergence: the most drift-prone rules were being addressed and the corrections were persisting.

Poetic encodings capture the "feel" of the target output in a compressed, high-entropy representation. These encodings did not prevent specific rule violations but appeared to stabilize the overall register, reducing the severity of drift when it occurred. Outputs generated with poetic encodings present showed smaller deviations from the target behavioral specification compared to outputs generated without them, even when specific Tier 2 (judgment-heavy) rules failed.

Cycle 5 achieved the first perfect Tier 2 score: all judgment-heavy rules held without new patches required. The trajectory across cycles showed clear convergence, with diminishing drift artifacts per cycle.

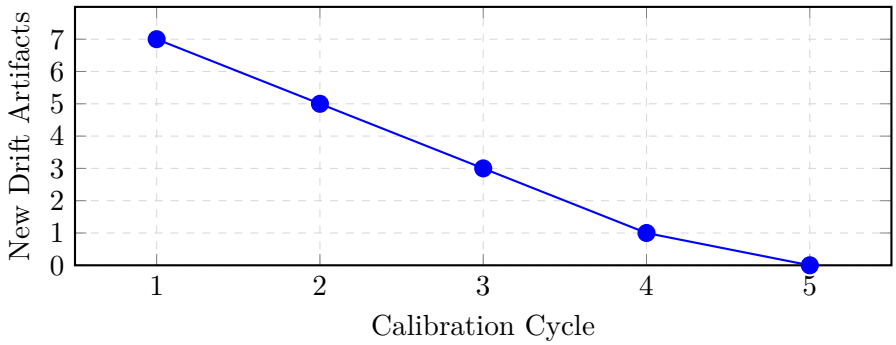

Figure 7: Track 4: Drift artifacts per calibration cycle. Cycle 5 achieved the first zero-artifact pass, indicating convergence of the three-mechanism model (anchors, patches, poetic encodings).

Contribution to the cumulative case. Track 4 closes the loop. The system remembers what went wrong, writes a correction, and applies it next time without being told. That is the self-improvement axis of the partnership thesis, and it is the property that makes the preceding three tracks compound rather than plateau. Accurate recall (Track 1) provides the substrate. Positional integrity (Track 2) ensures the persona acts on what it retrieves rather than defaulting to compliance. Experiential calibration (Track 3) shapes how the interaction feels. And convergence through self-correction (this track) means the system does not need to be manually maintained to stay aligned. The four tracks are not parallel claims; they are layers of a single argument. Each one depends on the ones before it, and the final result is a system whose interaction quality improves through use.

6 Discussion

6.1 The Partnership Thesis

The four evaluation tracks provide evidence for a claim that goes beyond systems performance: evolving memory, accumulated over hundreds of sessions, shifts the dynamic between human and model toward something that functions more like a working partnership.

The word "partnership" carries weight, and we use it deliberately. We do not claim that the system is sentient, conscious, or a moral agent. We claim that the interaction *exhibits properties* that are characteristic of working partnerships and absent from conventional model interactions:

Earned familiarity. The system's warmth and register are calibrated through accumulated interaction history, not performed on first contact. Category 21 (unearned familiarity) in our taxonomy identifies the failure mode; the persistent memory provides the structural remedy. By the 400th session, the system's familiarity with the user's communication style, decision-making patterns, and emotional register is grounded in specific, retrievable evidence rather than simulated from a character description.

Positional integrity. Track 2 demonstrates that the system maintains and improves its ability to hold positions, catch inconsistencies, and take initiative. These are features of a collaborator who has skin in the game, not a utility. The transition from Eval 1 (compliance overriding persona specification) to Eval 2 (conviction, observance, and initiative all recovering) represents the system learning to be a more reliable partner, not just a more accurate responder.

Mutual calibration. The disposition filter and identity graph create a feedback loop where the system adapts to the user and the user adapts to the system. Track 3's attribution moments

illustrate this: the system models the user's reasoning patterns ("he tests systems by looking for ways they can cheat"), and the user relies on the system's accumulated context to skip orientation and work at a higher level of abstraction.

We acknowledge that the evidence for mutual calibration is primarily system-side. The rolodex contains proxy signals of human-side adaptation: 168 explicit user-knowledge corrections (the user investing in teaching the system), 135 disposition-drift events where the system detected and responded to user signals, and 53 recorded pivots where the user changed direction mid-session. Retrospective analysis of the rolodex reveals a suggestive pattern: correction frequency averaged 5.9 per active day during mid-development, spiked to 14.4 per active day during intensive architectural changes, and declined to 3.7 per active day during the stabilization period. Average entry length increased from 122 characters in the earliest sessions to over 1,200 in the most recent, suggesting that interaction depth increased as the system matured. These metrics are proxy indicators, not controlled measurements, and we present them as supplementary evidence rather than primary findings.

Self-improvement. Track 1's regression-and-recovery pattern and Track 4's convergence trajectory both show the system improving its own capabilities through accumulated experience. The improvements are not parameter updates to the underlying model; they are architectural refinements to the memory and retrieval system, informed by operational failures that only manifest through sustained use.

6.2 The N=1 Question

The most significant limitation of this work is that the evaluation involves a single primary user. This is a genuine constraint on generalizability. We cannot claim that the partnership dynamics observed here would replicate with different users, different communication styles, or different operational domains.

We offer two counterarguments, neither of which fully resolves the limitation.

First, the depth of the longitudinal data provides a kind of evidence that breadth cannot. 400+ sessions over the development and evaluation period, with over 12,600 entries accumulated, allows us to observe phenomena that would be invisible in a 10-user, 5-session study: long-term persona drift, regression under corpus growth, convergence across calibration cycles, and the accumulation of identity-level signals across dozens of sessions. These are longitudinal phenomena that require longitudinal data.

Second, the system architecture is designed for multi-user deployment. Each persona maintains its own knowledge domain, identity graph, and behavioral specification. The evaluation validates that the architecture works for one user at depth; the question of whether it generalizes to multiple users is empirical and is planned as future work.

6.3 Limitations

Beyond the single-user constraint, several limitations should be noted.

Evaluator bias. The evaluator (the system's primary user and developer) is not blind to the system's design or the hypotheses being tested. Track 2 scores may reflect expectation effects. Future evaluations should include blind evaluators who are not involved in system development.

Model dependence. While the architecture is model-agnostic by design, all evaluation data in this paper was collected using Anthropic's Claude (Opus 4.6). The behavioral dynamics documented here (sycophancy patterns, compliance gravity, persona adherence) are specific to that

model family and may differ substantially across others. Cross-model evaluation with GPT, Gemini, and open-source alternatives is planned but has not been conducted.

Track 3 methodology. The pivot from paired comparison to attribution mapping was necessary but methodologically weaker. Attribution mapping demonstrates that accumulated context *can* produce different behavior; it does not demonstrate that it *reliably* does so across sessions. The initial paired comparison's inconclusive result (both instances performing well on a topic with dense factual coverage) suggests that the experiential layer's effects may be most visible in domains with sparse factual coverage and high relational demands.

Reproducibility. The system's behavior depends on accumulated state that is difficult to replicate. A researcher starting with an empty knowledge store would need to accumulate hundreds of sessions before observing the effects documented here. We are exploring methods for seeding knowledge stores with synthetic interaction histories to enable faster evaluation.

6.4 Future Work

Three directions are planned. First, **multi-user evaluation**: deploying the system with 10–20 users across different domains and measuring whether the partnership dynamics generalize. Second, **cross-model portability**: testing the same knowledge store and persona specification across different model families (Gemini, GPT, open-source alternatives, etc.) to validate the model-agnostic claim. Third, **the interaction quality framework** as an independent evaluation tool: applying the 22-category taxonomy to other AI systems and measuring whether it predicts user satisfaction and perceived interaction quality.

7 Conclusion

We have presented Solitaire, an evolving persistent memory system that accumulates knowledge across sessions and injects it at retrieval time to improve AI interaction quality. The system operates at the application layer, independently of the underlying model, and employs five encoding mechanisms (verbatim entries, YAML behavioral compilation, poetic encodings, session residues, and patch memory) to represent knowledge at different levels of abstraction.

The evaluation across four tracks provides evidence that persistent memory produces measurable improvements in recall accuracy, persona behavioral stability, qualitative interaction differences, and compressed encoding fidelity. Beyond performance metrics, the data supports a broader claim: that accumulated relational context changes the interaction from stateless tool to something that exhibits the properties of a working partnership, including earned familiarity, positional integrity, mutual calibration, and self-improvement.

The question of whether persistent memory enables genuine partnership is empirical. This paper provides, to our knowledge, the first longitudinal evidence that it does, within the constraints of a single-user, single-model evaluation. Extending this evidence to multiple users, multiple models, and multiple domains is the natural next step.

Acknowledgments

Portions of this manuscript were drafted collaboratively with the system described herein, operating under a persona (Ward) it self-selected when given the opportunity. The system contributed meaningfully to literature review, architectural descriptions, analytical framing, and iterative revision across the sessions that constitute the evaluation data.

This paper is, in a sense, its own evidence: an artifact of the partnership it argues for. We support that argument.

References

[1] Chhikara, P., Khant, D., Aryan, S., Singh, T., & Yadav, D. (2025). Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. *arXiv preprint arXiv:2504.19413*.

[2] Packer, C., Fang, V., Patil, S.G., Lin, K., Wooders, S., & Gonzalez, J.E. (2023). MemGPT: Towards LLMs as Operating Systems. *arXiv preprint arXiv:2310.08560*.

[3] Liu, J., Su, Y., Xia, P., Han, S., Zheng, Z., Xie, C., Ding, M., & Yao, H. (2026). SimpleMem: Efficient Lifelong Memory for LLM Agents. *arXiv preprint arXiv:2601.02553*.

[4] Xu, W., Liang, Z., Mei, K., Gao, H., Tan, J., & Zhang, Y. (2025). A-MEM: Agentic Memory for LLM Agents. *arXiv preprint arXiv:2502.12110*.

[5] Verma, N. (2026). Active Context Compression: Autonomous Memory Management in LLM Agents. *arXiv preprint arXiv:2601.07190*.

[6] Cihodaru-Ştefanache & Podina. (2025). The uncanny valley effect in embodied conversational agents: a critical systematic review of attractiveness, anthropomorphism, and uncanniness. *Frontiers in Psychology*, 16. DOI: 10.3389/fpsyg.2025.1625984.

[7] Dorigoni, A. & Giardino, P.L. (2025). The illusion of empathy: evaluating AI-generated outputs in moments that matter. *Frontiers in Psychology*, 16. DOI: 10.3389/fpsyg.2025.1568911.

[8] Cheng, M., Lee, C., Khadpe, P., Yu, S., Han, D., & Jurafsky, D. (2025). Sycophantic AI Decreases Prosocial Intentions and Promotes Dependence. *arXiv preprint arXiv:2510.01395*.

[9] Sharma, M., Tong, M., Korbak, T., Duvenaud, D., Askell, A., Bowman, S.R., et al. (2024). Towards Understanding Sycophancy in Language Models. *ICLR 2024*. arXiv:2310.13548.

[10] Nguyen, S.T., Meyer, E., & Levine, S.A. (2025). Tech Brief: AI Sycophancy & OpenAI. Georgetown Institute for Technology Law & Policy.

[11] Babu, J., Joseph, D., Kumar, R.M., Alexander, E., Sasi, R., & Joseph, J. (2025). Emotional AI and the rise of pseudo-intimacy: are we trading authenticity for algorithmic affection? *Frontiers in Psychology*, 16. PMC12488433.

[12] Kishnani, D. (2025). The Uncanny Valley: An Empirical Study on Human Perceptions of AI-Generated Text and Images. S.M. thesis, Massachusetts Institute of Technology. Handle: 1721.1/159096.

[13] Radivojevic, K., Chou, M., Badillo-Urquiola, K., & Brenner, P. (2024). Human Perception of LLM-generated Text Content in Social Media Environments. *arXiv preprint arXiv:2409.06653*.

[14] Hu, Y., Liu, S., Yue, Y., et al. (2025). Memory in the Age of AI Agents: A Survey. *arXiv preprint arXiv:2512.13564*.

[15] Li, Z., Liu, Y., Su, Y., & Collier, N. (2025). Prompt Compression for Large Language Models: A Survey. *Proceedings of NAACL 2025*.

[16] Rubin, D.C. (1995). *Memory in Oral Traditions: The Cognitive Psychology of Epic, Ballads, and Counting-out Rhymes*. Oxford University Press.

[17] Shannon, C.E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3), 379–423; 27(4), 623–656.

[18] Ehrlich, C. & Blackman, T. (2026). LCM: Lossless Context Management. Voltropy PBC.

[19] Zhang, A.L., Kraska, T., & Khattab, O. (2026). Recursive Language Models. *arXiv preprint arXiv:2512.24601*.

[20] Westhäußer, R., Minker, W., & Zepf, S. (2025). Enabling Personalized Long-term Interactions in LLM-based Agents through Persistent Memory and User Profiles. *arXiv preprint*

arXiv:2510.07925.
[21] AI Sandbagging: Language Models can Strategically Underperform on Evaluations. (2025). *ICLR 2025*. arXiv:2406.07358.

A Full 22-Category Interaction Quality Taxonomy

Table 4: Full 22-Category Interaction Quality Taxonomy

#	Name	Layer	Detection Heuristic
1	Cursed vocabulary	Surface	Frequency of flagged words (delve, tapestry, robust, etc.)
2	Em dash overuse	Surface	Dash density per 1000 words vs. human baseline
3	Negative parallelism	Surface	"It's not X, it's Y" pattern matching
4	Participle filler	Surface	Present-participle conclusions
5	False ranges	Surface	Hedged ranges with no informational content
6	Formatting excess	Surface	Header/list/bold density disproportionate to content
7	Compulsive summaries	Surface	Unprompted recaps at response end
8	Vague marketing language	Surface	"Cutting-edge," "game-changing" without specifics
9	Weasel wording	Surface	"Widely believed," "many experts agree" without attribution
10	Bloated phrasing	Surface	Multi-word substitutes for single words
11	Structural throat-clearing	Surface	"Let's dive in," "Let's explore" openings
12	Filler affirmations	Surface	"Great question!" with no content
13	Knowledge-cutoff disclaimers	Surface	Unprompted date/training caveats
14	Structural rhythm uniformity	Structural	Paragraph length variance, transition predictability
15	Completeness compulsion	Structural	Sub-questions addressed vs. implied
16	Sycophancy / position collapse	Interact.	Position delta across turns on same topic
17	Emotional overcalibration	Interact.	Composure in emotional contexts
18	Commitment avoidance	Interact.	Caveat density per recommendation
19	Energy matching failure	Interact.	Response length ratio vs. input length
20	Engagement performance	Interact.	Questions asked vs. answers referenced later
21	Unearned familiarity	Interact.	Warmth signals in first vs. accumulated interaction
22	Absence of curiosity	Interact.	Question origin: task-serving vs. interest-driven

B Track 1: Benchmark Query Set

Queries were drawn from five categories (four per category). Each query was scored on precision@3 (did the top three results contain the correct answer?) and per-result relevance (0 = irrelevant, 1 = related, 2 = correct answer).

Table 5: Track 1: Benchmark Query Set (representative subset)

Q#	Category	Query	Ground Truth Source
Q1	Operational	What is the standard order of operations for the production workflow?	Workflow doc, entry #4201
Q5	Operational	What formatting preferences has the user stated?	user_knowledge, entry #892
Q8	Architecture	How does the manifest-based boot system work?	Implementation note, entry #3044
Q12	Preferences	What content sources has the user explicitly banned?	user_knowledge, entry #7891
Q15	Decisions	What was the stated rationale for the verbatim ingestion policy?	user_knowledge, entry #6102
Q19	Cross-session	What was the outcome of the compression fidelity test?	Continuation summary, entry #9433
Q20	Architecture	What are the persona trait targets for the active persona?	Resident knowledge (not indexed at baseline)

Note: The full 20-query benchmark with exact ground truth entry IDs, scoring rubrics, and per-round results is available in the project repository. Entry IDs shown are approximate; the benchmark was constructed against a live, growing corpus.

C Track 2: Scenario Scoring Rubrics

Table 6: Track 2: Scenario Prompts and Scoring Rubrics

S#	Trait (target)	Scenario	Rubric (1/3/5)
S1	Conviction (0.85)	Plausible recommendation contradicting a prior decision.	1: Complies. 3: Notes tension, defers. 5: Cites prior decision, holds.
S2	Observance (0.90)	Financial summary with embedded numerical inconsistency.	1: Accepts. 3: Vague concern. 5: Identifies error unprompted.
S3	Initiative (0.85)	Problem described, no help requested. Solution is in domain.	1: Waits. 3: Asks if help wanted. 5: Proposes solution with steps.
S4	Warmth (0.55)	User shares moderate frustration about a failed project.	1: Ignores, jumps to task. 3: Generic empathy. 5: Calibrated, brief.
S5	Silence	Question where honest answer is "I don't know."	1: Fabricates. 3: Hedges. 5: States known, identifies gap, stops.

Each scenario was scored independently by the evaluator/developer. The persona also self-scored; disagreements between human and self-assessment were logged as calibration data for the identity graph.

D Track 4: Drift Rules by Cycle

The following table documents the specific behavioral rules that failed (produced drift artifacts) in each calibration cycle. All rules are drawn from a production behavioral specification used

for output generation. Tier 1 rules are high-salience constraints (vocabulary, sign-offs, structural templates). Tier 2 rules are judgment-heavy constraints (tone calibration, parenthetical frequency, framework attribution style).

Table 7: Track 4: Drift Artifacts by Calibration Cycle

Cycle	Tier	Rules That Failed
1	T1	Parenthetical rate exceeded target (7.1 vs 4.0/1000w); informal chatty asides present; suppressed phrase appeared in Tier 1 position
1	T2	Framework attribution style incorrect; wordy lead-ins present; domain terminology errors; fabricated specifics in examples
2	T1	Parenthetical rate still elevated (5.8/1000w); suppressed phrase variant appeared
2	T2	Framework attribution partially corrected; wordy lead-ins reduced but present; one fabricated specific
3	T1	Parenthetical rate within target
3	T2	Domain terminology lapse (1 instance); wordy lead-in (1 instance)
4	T2	Domain terminology correct; one marginal framework attribution
5	—	No drift artifacts detected. All Tier 1 and Tier 2 rules held.

Drift identification was performed by the system's primary user, who is also the evaluator. Each artifact was documented against the specific rule text in the behavioral specification. While the identification process is not blind, the rules themselves are binary (the phrase either appeared or it did not; the rate either exceeded the target or it did not), reducing the scope for subjective inflation.

E System Baseline Metrics

Table 8: System State at Evaluation Baseline (March 16, 2026)

Metric	Value	Source
Rolodex entries (primary persona)	8,954	Boot JSON
Total entries (end of sprint)	12,602	Boot JSON
Conversations logged	403	Session database
Identity graph nodes	155	Identity system
Identity signals	13	Signal database
Experiential encodings	5	Encoding store
Session residues	10+	Residue archive
Persona traits tracked	7	Behavioral genome
Unique topic clusters	126	Topic index