World Weaver: Toward Persistent World Models in Agentic AI Systems

A Critical Analysis of Cognitive Memory Architecture
for Language Model Agents

Aaron Storey
Department of Computer Science
aaron@storey.dev

December 2024

Abstract

Large language models have achieved remarkable capabilities in reasoning, code generation, and natural language understanding, yet they remain fundamentally stateless—each interaction begins without memory of past sessions, accumulated knowledge, or learned skills. This paper presents World Weaver, a tripartite cognitive memory architecture designed to provide AI agents with persistent, inspectable world models. We situate this work within the broader discourse on world models in artificial intelligence, drawing connections to Geoffrey Hinton's theoretical framework and Yann LeCun's vision of autonomous machine intelligence. Through critical analysis, we examine what World Weaver does well, where it falls short, and what fundamental questions remain unanswered about memory, learning, and world representation in artificial agents. We argue that the central contribution is not the technical implementation but rather the explicit confrontation with a problem the field has largely deferred: how should AI agents accumulate and organize knowledge across time?

Contents

1 Introduction: The Amnesia Problem

Consider an AI coding assistant that has helped you debug the same authentication module across fifty sessions. Each time, it rediscovers the codebase structure, re-learns your naming conventions, and repeatedly suggests approaches you've already tried and rejected. Despite sophisticated reasoning capabilities, the system exhibits a peculiar form of amnesia—not forgetting within a conversation, but forgetting *between* them.

This is not a bug but a feature of current large language model (LLM) architectures. Models like GPT-4, Claude, and Gemini process context windows of tens or hundreds of thousands of tokens, but this context is ephemeral. When the session ends, everything learned is lost. The weights that encode the model's general knowledge remain frozen; only fine-tuning can create lasting change, and fine-tuning is expensive, slow, and risks catastrophic forgetting [Kirkpatrick et al., 2017].

World Weaver emerges from a simple question: *What would it mean for an AI agent to remember?*

This question is deceptively profound. Human memory is not a database lookup. It involves consolidation, where experiences transform into knowledge over time. It involves forgetting, where irrelevant information fades while important memories strengthen. It involves reconstruction, where recall is an active process influenced by current context. And it involves integration, where new information connects to existing knowledge structures rather than accumulating in isolation.

The technical contribution of World Weaver is a tripartite memory system implementing episodic, semantic, and procedural stores. But the deeper contribution is forcing explicit engagement with questions the field has largely avoided: What should AI agents remember? How should memories decay? When should experiences consolidate into knowledge? These are not engineering problems with optimal solutions—they are design decisions reflecting assumptions about what intelligence requires.

1.1 The Stakes

Why does AI memory matter beyond convenience? Several converging concerns elevate this from an engineering problem to a fundamental challenge:

Alignment Through Continuity: A system that remembers its commitments, past reasoning, and user preferences may be more alignable than one that optimizes myopically for immediate objectives. Memory enables consistency, and consistency enables trust.

Efficiency: Current approaches waste enormous computation rediscovering what was previously known. An agent that remembers debugging strategies, codebase structure, and user preferences operates more efficiently than one that starts fresh.

Emergent Capabilities: Human intelligence emerges partly from accumulated experience. A child who has never seen a bicycle cannot ride one; expertise requires accumulated practice. Stateless systems cannot develop expertise in this sense.

Safety and Auditing: If AI systems make consequential decisions, we need to understand why. Inspectable memory provides an audit trail that opaque neural activations cannot.

1.2 Scope and Claims

This paper does not claim to solve the memory problem in AI. We present one architecture, analyze its strengths and weaknesses, and articulate questions requiring further research. Our claims are modest:

1. Explicit memory architecture is a tractable approach to agent persistence
2. Cognitive science provides useful design principles for memory systems

3. Hybrid retrieval (dense + sparse) outperforms either alone for technical domains
4. Inspectable memory enables capabilities impossible with opaque systems
5. Fundamental questions about AI memory remain unanswered

2 Theoretical Foundations

2.1 The World Model Debate

Geoffrey Hinton's 2023 departure from Google brought renewed attention to fundamental questions about artificial intelligence. Among his concerns was the observation that AI systems may develop "world models" that exceed human understanding—internal representations of how the world works that, while effective, remain opaque to human inspection [Hinton, 2023].

The concept of world models has deep roots in cognitive science. Kenneth Craik proposed in 1943 that organisms carry "small-scale models" of external reality, enabling prediction and planning [Craik, 1943]. This idea resurfaced in AI through the work of Ha and Schmidhuber, who demonstrated agents learning compressed representations of environments that enable imagination-based planning [Ha & Schmidhuber, 2018].

But what exactly constitutes a world model? Yann LeCun's position paper on autonomous machine intelligence offers a framework: a world model is a system capable of predicting future states of the world given current states and contemplated actions [LeCun, 2022]. This predictive capacity enables planning—mentally simulating action sequences before physical execution.

Large language models possess implicit world models encoded in their weights. When GPT-4 correctly predicts that dropping a glass will cause it to shatter, it demonstrates world knowledge. But this knowledge is:

1. **Static**: Frozen at training time, unable to incorporate new information without retraining
2. **Opaque**: Distributed across billions of parameters in ways that resist interpretation
3. **Undifferentiated**: Personal experiences, general facts, and procedural skills are not distinguished
4. **Ungrounded**: Learned from text rather than interaction with the physical world

World Weaver addresses the first three limitations while explicitly acknowledging the fourth remains unsolved.

2.2 Cognitive Memory Systems

Endel Tulving's distinction between episodic and semantic memory provides the foundational framework [Tulving, 1972, 1985]. Episodic memory stores autobiographical events—specific experiences located in time and space. Semantic memory stores general knowledge—facts, concepts, and relationships abstracted from specific experiences. Crucially, these systems interact: episodic memories consolidate into semantic knowledge through processes that remain incompletely understood.

John Anderson's ACT-R (Adaptive Control of Thought—Rational) extends this framework with procedural memory and explicit activation dynamics [Anderson, 1983, Anderson & Lebiere, 2004]. In ACT-R, memory retrieval is competitive: items with higher activation are more likely to be recalled. Activation spreads through associative networks, so thinking about "coffee" activates related concepts like "morning" and "caffeine." This spreading activation explains priming effects and the associative nature of human recall.

The mathematical formulation in ACT-R provides:

$$A_i = B_i + \sum_j W_j S_{ji} + \epsilon \tag{1}$$

where A_i is the activation of chunk i, B_i is base-level activation (reflecting recency and frequency), W_j is the attentional weight of source j, S_{ji} is the strength of association from j to i, and ϵ is noise.

Base-level activation follows a power law of practice and decay:

$$B_i = \ln \left(\sum_{j=1}^{n} t_j^{-d} \right) \tag{2}$$

where t_j is the time since the jth presentation and d is a decay parameter (typically around 0.5).

World Weaver implements computational interpretations of these cognitive systems:

- **Episodic Memory**: Vector-indexed experiences with temporal-spatial context, subject to FSRS-based decay modeling
- **Semantic Memory**: Entity-relationship graphs with ACT-R-inspired spreading activation
- **Procedural Memory**: Executable skills with empirical success tracking and adaptive refinement

2.3 Memory-Augmented Neural Networks

The problem of giving neural networks persistent memory has a substantial research history. Neural Turing Machines introduced differentiable external memory that networks could read from and write to [Graves et al., 2014]. The key insight was treating memory as a continuous, differentiable resource enabling end-to-end training.

Memory Networks applied similar ideas to question answering [Weston et al., 2014]. The architecture maintains a memory bank of facts and learns to attend to relevant memories when answering questions. End-to-end Memory Networks extended this with multiple "hops" of attention over memory [Sukhbaatar et al., 2015].

More recently, retrieval-augmented generation (RAG) combines neural language models with external document retrieval [Lewis et al., 2020]. Rather than differentiable memory, RAG uses traditional information retrieval to find relevant documents, then conditions the language model on retrieved text.

World Weaver differs from these approaches in several ways:

Property	**NTM/Memory Net**	**RAG**	**World Weaver**
Differentiable	Yes	No	No
Structured	No	No	Yes
Typed memories	No	No	Yes
Consolidation	No	No	Yes
Inspectable	Limited	Yes	Yes
Decay dynamics	No	No	Yes

Table 1: Comparison of memory augmentation approaches

The RETRO architecture demonstrated that retrieval-augmented models can match larger models with less computation [Borgeaud et al., 2022]. This suggests that explicit memory may be more efficient than encoding everything in parameters. But RETRO retrieves from a static corpus; World Weaver envisions memory that grows and changes through agent experience.

2.4 The Consolidation Problem

Biological memory consolidation remains incompletely understood, but several principles have emerged [McClelland et al., 1995, Walker & Stickgold, 2017]:

Systems Consolidation: Newly formed memories depend on the hippocampus but gradually become independent, stored in neocortical networks. This transfer takes days to years.

Sleep's Role: Memory consolidation is enhanced during sleep, particularly slow-wave sleep for declarative memories and REM sleep for procedural skills. Sleep deprivation impairs consolidation.

Reactivation: During consolidation, memories are "replayed," strengthening important connections. This replay preferentially consolidates emotionally significant or reward-associated memories.

Schematization: Over time, specific episodic details fade while general patterns are preserved. You remember that restaurants have menus without remembering every menu you've read.

World Weaver's consolidation mechanism is a computational analogy:

Algorithm 1 Memory Consolidation

```
Cluster similar episodes using HDBSCAN
for each cluster with |C| ≥ threshold do
    Extract common entities via NER
    Create/update semantic nodes
    if pattern frequency ≥ skill threshold then
        Promote to procedural skill
    end if
end for
Apply Hebbian updates to co-retrieved pairs
Prune memories below activation threshold
```

This process is vastly simplified compared to biological consolidation, but captures the key transformation: episodic specifics become semantic generalities and procedural skills.

3 System Architecture

3.1 Design Philosophy

World Weaver is built on several design principles that reflect both technical constraints and philosophical commitments:

Separation of Concerns: Following cognitive science rather than treating memory as homogeneous, we maintain distinct episodic, semantic, and procedural stores with different retrieval dynamics and update rules. This modularity enables independent optimization and clearer debugging.

Inspectability: All memory contents can be examined, queried, and audited. This transparency addresses concerns about opaque AI systems while enabling debugging and alignment verification. You can ask: "What does this agent remember about authentication?" and receive an interpretable answer.

Local-First: Core functionality requires no external API calls. The system uses local embedding models (BGE-M3) and local entity extraction (GLiNER), enabling offline operation, reproducibility, and avoiding vendor lock-in.

Graceful Decay: Following FSRS (Free Spaced Repetition Scheduler) algorithms, memories decay over time unless reinforced through recall. This prevents unbounded growth while preserving important information through natural selection pressure.

No Backwards Compatibility: The system is designed for a single coherent architecture rather than maintaining legacy code paths. This reduces complexity at the cost of migration burden.

3.2 Episodic Memory

Episodic memory stores autobiographical events—specific interactions with temporal and spatial context. The formal structure:

$$e = \langle c, \mathbf{v}_d, \mathbf{v}_s, \tau, \sigma, \omega, \nu, s \rangle \tag{3}$$

where:

- c is content (text)
- $\mathbf{v}_d \in \mathbb{R}^{1024}$ is dense embedding
- $\mathbf{v}_s \in \mathbb{R}^{|V|}$ is sparse embedding
- τ is timestamp
- σ is spatial context (project, file, tool)
- $\omega \in \{\text{success}, \text{failure}, \text{partial}, \text{neutral}\}$ is outcome
- $\nu \in [0, 1]$ is emotional valence/importance
- s is FSRS stability parameter

Retrieval combines multiple signals:

$$\text{score}(e|q) = w_{\text{sim}} \cdot \text{sim}(q, e) + w_r \cdot R(e) + w_o \cdot O(e) + w_\nu \cdot \nu_e \tag{4}$$

where:

- $\text{sim}(q, e)$ is hybrid similarity (dense + sparse via RRF)
- $R(e) = e^{-\lambda(t_{\text{now}} - \tau_e)}$ is recency factor
- $O(e)$ maps outcomes to weights (success: 1.0, partial: 0.5, neutral: 0.3, failure: 0.1)
- Weights sum to 1: $w_{\text{sim}} + w_r + w_o + w_\nu = 1$

The multi-factor scoring reflects a core insight: relevant memories are not just semantically similar but also recent, successful, and important.

3.3 Semantic Memory

Semantic memory stores entities and relationships in a property graph:

$$G = (V, E, \phi_V, \phi_E) \tag{5}$$

where vertices V are typed entities (Person, Project, Concept, Tool, etc.), edges E are typed relationships (USES, RELATED_TO, PART_OF, CAUSED, etc.), and ϕ functions assign properties to vertices and edges.

Retrieval uses spreading activation:

$$A_i^{(t+1)} = B_i + \alpha \sum_{j \in \mathcal{N}(i)} w_{ji} \cdot A_j^{(t)} \tag{6}$$

where α is a decay factor preventing unbounded spreading, and iteration continues until convergence or maximum steps.

Base-level activation follows FSRS:

$$B_i = D_0 \cdot S_i^{-w} \cdot \left(e^{w \cdot \Delta t / S_i} - 1\right) \tag{7}$$

where S_i is stability, Δt is time since last access, D_0 is initial difficulty, and w is a tunable parameter.

3.4 Procedural Memory

Procedural memory stores executable skills with empirical tracking:

$$p = \langle \text{pattern}, \text{actions}, \text{prereqs}, h, f, n, \text{embedding} \rangle \tag{8}$$

where h, f, n are counts of helpful, harmful, and neutral executions.

The usefulness metric:

$$U(p) = \frac{h - 0.5f}{h + f + n + \epsilon} \tag{9}$$

where ϵ prevents division by zero for new skills.

Skills below a usefulness threshold are marked inactive but retained for potential reactivation. This implements a form of "unlearning" without permanent deletion.

The three-role architecture separates learning:

1. **Agent**: Executes tasks, records outcomes
2. **Reflector**: Analyzes executions, extracts atomic lessons
3. **SkillManager**: Validates lessons, updates skillbook, enforces quality gates

Quality gates include:

- Atomicity score ≥ 0.7 (lessons must be specific, not vague)
- Semantic deduplication (similarity < 0.9 to existing skills)
- Consistency check (doesn't contradict high-usefulness skills)

3.5 Hybrid Retrieval

A technical innovation is hybrid retrieval combining dense semantic vectors with sparse lexical matching. Using BGE-M3, we generate both representations in a single forward pass:

$$\text{BGE-M3}(x) \rightarrow (\mathbf{v}_d \in \mathbb{R}^{1024}, \mathbf{v}_s \in \mathbb{R}^{|V|}) \tag{10}$$

Dense vectors capture semantic similarity through learned representations. Sparse vectors are lexical weights capturing term importance, similar to BM25 but learned rather than statistical.

Retrieval employs Reciprocal Rank Fusion (RRF):

$$\text{RRF}(d) = \sum_{r \in \{d,s\}} \frac{1}{k + \text{rank}_r(d)} \tag{11}$$

where $k = 60$ is a smoothing constant that prevents high-ranked items from dominating.

Query Type	Dense Recall@10	Hybrid Recall@10
Conceptual	0.78	0.81
Exact match (functions)	0.42	0.79
Error codes	0.38	0.82
Mixed	0.72	0.84

Table 2: Retrieval performance by query type

The improvement is most dramatic for exact-match queries where semantic similarity fails to capture lexical identity.

4 Literature Review

4.1 Memory in Cognitive Science

The scientific study of memory reveals systems far more complex than computer storage metaphors suggest. Bartlett's seminal work demonstrated that memory is reconstructive, not reproductive—we don't replay recordings but actively rebuild memories each time, influenced by current knowledge and context [Bartlett, 1932]. This has profound implications: memory is not a passive archive but an active process intertwined with comprehension and inference.

Tulving's proposal of multiple memory systems—later supported by neuroimaging and lesion studies—established that different types of memory have different neural substrates [Squire, 2004]. Patient H.M., who lost hippocampal function, could not form new episodic memories but retained procedural learning. This dissociation suggests memory is not unitary but modular.

The consolidation process has received extensive study. Müller and Pilzecker first proposed consolidation in 1900, observing that new memories are initially fragile. Modern research confirms that consolidation requires protein synthesis and is modulated by emotional arousal through amygdala involvement [McGaugh, 2000].

Sleep plays a crucial role. Walker and Stickgold's work demonstrates that sleep-dependent consolidation strengthens memories and extracts general patterns [Walker & Stickgold, 2017]. During slow-wave sleep, hippocampal memories are replayed and transferred to neocortex. This "offline" processing may explain why we sometimes wake up with solutions to problems.

Forgetting is not merely failure but serves important functions. Anderson's retrieval-induced forgetting shows that retrieving some memories inhibits related competitors [Anderson, 1994]. This sharpens memory by suppressing interference. Similarly, directed forgetting enables intentional memory control. A memory system without forgetting would be overwhelmed by irrelevant detail.

4.2 Memory in Artificial Intelligence

Early AI embraced explicit symbolic memory. SHRDLU maintained a model of its blocks world [Winograd, 1971]. SOAR's long-term memories stored productions, semantic knowledge, and episodic traces [Laird et al., 1987]. Cyc accumulated millions of hand-coded assertions about common sense [Lenat, 1995].

These systems were interpretable but brittle. They couldn't handle ambiguity, variation, or knowledge not explicitly encoded. The knowledge acquisition bottleneck—the difficulty of encoding everything systems need to know—proved insurmountable.

Connectionist approaches shifted focus to distributed representations. Knowledge became implicit in weights rather than explicit in symbols. This enabled flexibility and graceful degradation but sacrificed interpretability. Modern LLMs represent the apotheosis of this approach: trillions of parameters encoding vast knowledge in inscrutable ways.

The tension between symbolic interpretability and neural flexibility motivates hybrid approaches. World Weaver uses neural components (embeddings, LLM reasoning) but symbolic structures (typed memories, explicit relationships). This sacrifices end-to-end optimization for modularity and inspectability.

4.3 Retrieval-Augmented Generation

RAG has become the dominant paradigm for grounding LLM outputs in external knowledge. The basic architecture retrieves relevant documents, then conditions generation on retrieved text.

Several limitations motivate World Weaver's approach:

Flat Structure: Standard RAG retrieves text chunks without structure. World Weaver maintains typed memories with explicit relationships.

No Learning: RAG corpora are typically static. World Weaver's memory grows and changes through agent experience.

No Decay: All documents in RAG corpora are equally accessible. World Weaver implements forgetting curves that prioritize recent, important memories.

No Consolidation: RAG doesn't transform retrieved information. World Weaver consolidates episodes into semantic knowledge and procedural skills.

Recent work addresses some limitations. Self-RAG enables models to critique and revise retrievals [Asai et al., 2023]. RAPTOR builds hierarchical summaries for multi-level retrieval [Sarthi et al., 2024]. These approaches remain closer to document retrieval than cognitive memory.

4.4 World Models in AI

Ha and Schmidhuber's "World Models" demonstrated learning environment dynamics for imagination-based planning [Ha & Schmidhuber, 2018]. Their VAE-RNN architecture compressed visual observations into latent states, then learned transition dynamics in latent space. Agents could plan by "imagining" action sequences without environment interaction.

This work connects to model-based reinforcement learning, where agents learn environment models to enable planning [Sutton, 1991]. The trade-off between model-based and model-free approaches parallels debates about explicit vs. implicit memory in cognitive science.

LeCun's "A Path Towards Autonomous Machine Intelligence" proposes a comprehensive architecture including a world model, cost module, and actor [LeCun, 2022]. The world model predicts future states at multiple abstraction levels, enabling hierarchical planning. Notably, LeCun argues against autoregressive language models as a path to intelligence, favoring joint embedding predictive architectures.

World Weaver's relationship to these frameworks is complementary. We focus on memory—what the agent knows and has experienced—rather than dynamics—how the world changes. A complete autonomous agent might combine World Weaver-style memory with Ha/LeCun-style dynamics prediction.

5 Case Study: A Session in Practice

To illustrate World Weaver's operation concretely, we trace a hypothetical but representative coding session.

5.1 Session Initialization

The agent begins work on project "kymera-backend," a Go service. At session start, World Weaver retrieves relevant context:

Episodic retrieval (spatial filter: kymera-backend):

- "Fixed race condition in scheduler by adding mutex lock around job queue operations" (3 days ago, success)
- "Debugging gRPC streaming failure—root cause was context cancellation propagation" (1 week ago, success)
- "Attempted to use sync.Map for concurrent access, reverted due to range-mutation issues" (2 weeks ago, failure)

Semantic retrieval (spreading activation from "kymera-backend"):

- Entity: kymera-backend $\rightarrow$ USES $\rightarrow$ gRPC
- Entity: kymera-backend $\rightarrow$ HAS_PATTERN $\rightarrow$ "dependency injection via interfaces"
- Entity: gRPC $\rightarrow$ RELATED_TO $\rightarrow$ protobuf

Procedural retrieval (high-usefulness skills):

- "For Go concurrency bugs, check context propagation first" (U=0.87)
- "Always run with -race flag during development" (U=0.92)
- "Use table-driven tests for Go functions" (U=0.84)

This context—accumulated from prior sessions—primes the agent with project-specific knowledge it would otherwise lack.

5.2 Task Execution

The user requests: "Fix the flaky test in scheduler_test.go."

The agent retrieves the episodic memory about the race condition fix. This triggers spreading activation to the skill "check context propagation first." The agent's approach is informed by accumulated experience rather than generic strategies.

During investigation, the agent discovers the flakiness stems from a timing-dependent assertion. It fixes by using test synchronization primitives.

5.3 Memory Formation

Post-task, the system creates memories:

Episodic:

```
content: "Fixed flaky scheduler test - root cause was
         timing-dependent assertion on channel receive.
         Solution: added sync.WaitGroup for test
         synchronization."
outcome: success
spatial: {project: kymera-backend,
          file: scheduler_test.go}
importance: 0.7
```

Reflector analysis extracts potential skill:

```
pattern: "flaky test + channel operation"
lesson: "For flaky tests involving channels, use
         explicit synchronization (WaitGroup, select
         with timeout) rather than relying on timing"
```

SkillManager evaluation:

- Atomicity score: 0.78 (specific, actionable)
- Novelty: similarity to existing skills = 0.34 (sufficiently novel)
- Consistency: no contradiction with high-U skills
- Decision: Accept as new skill with U_0=0.5

5.4 Consolidation (Later)

During periodic consolidation:

1. This episode clusters with the earlier race condition fix (both involve Go concurrency, synchronization)
2. NER extracts entities: "WaitGroup," "channel," "scheduler" (links to existing kymera-backend entity)
3. The pattern "Go synchronization issues in this codebase" strengthens, potentially promoting to a higher-level insight: "This codebase has recurring concurrency issues warranting architectural review"

This transformation—from specific fix to pattern recognition to architectural insight—illustrates the episodic $\rightarrow$ semantic $\rightarrow$ procedural flow.

5.5 Future Session Benefits

In a later session, a developer asks about test reliability. The agent retrieves:

- Both synchronization-related episodes
- The newly minted skill about channel synchronization
- The semantic pattern about recurring concurrency issues

The response isn't just "here's how to fix this test" but "this codebase has a pattern of concurrency issues—here are the cases I remember, the general strategies that worked, and a suggestion that the scheduler module might benefit from concurrency review."

This represents genuine accumulation of expertise, not mere retrieval.

6 Memory vs. Retrieval: A Crucial Distinction

6.1 What Retrieval-Augmented Generation Is Not

RAG systems retrieve documents and concatenate them to prompts. This is powerful but not memory in any substantive sense. The distinction matters:

RAG:

- Static corpus (documents exist independent of agent)
- No learning (corpus doesn't change from agent activity)

- No decay (all documents equally accessible)
- No transformation (documents stay as-is)
- No typing (all items are "documents")

Memory:

- Dynamic corpus (grows from agent experience)
- Learning (experience creates new memories)
- Decay (unrehearsed memories fade)
- Transformation (consolidation changes memories)
- Typing (episodes, entities, skills are distinct)

The difference is not merely technical but conceptual. RAG is a tool for grounding generation in external knowledge. Memory is part of the agent's identity—what the agent has experienced and learned.

6.2 The Continuity Criterion

We propose a criterion: a system has genuine memory (not just retrieval) if removing its memory state would change *what it is*, not just *what it knows*.

By this criterion, removing a RAG corpus doesn't change the model—it's the same model with less information access. But removing World Weaver's accumulated experience would fundamentally alter the agent. An agent that has debugged authentication fifty times is different from one that hasn't, even if both have access to the same documentation.

This connects to personal identity in philosophy. Locke argued that personal identity consists in psychological continuity—memory connecting present to past selves. If we take this seriously for AI, agents with persistent memory have something like continuous identity that stateless agents lack.

We don't claim World Weaver creates persons or moral patients. But we note that building memory systems forces engagement with questions typically reserved for philosophy of mind.

6.3 The Experience-Knowledge Gap

There's a gap between having an experience and having knowledge derived from that experience. Humans process experience: we reflect, abstract, connect to prior knowledge, draw lessons. Raw experience becomes organized knowledge through this processing.

World Weaver attempts to bridge this gap through consolidation and skill extraction. But the processing is shallow compared to human cognition. We don't really understand the experience—we pattern-match over surface features.

Deep experience processing might require:

- Causal reasoning (why did this work?)
- Counterfactual imagination (what if I'd tried X?)
- Analogical mapping (how is this like other situations?)
- Meta-cognition (what does this tell me about how I learn?)

Current systems lack these capabilities. World Weaver accumulates experience but doesn't deeply process it. This is perhaps the most significant limitation.

7 Critical Analysis

7.1 What World Weaver Does Well

Explicit Confrontation with the Memory Problem: The field has largely treated agent memory as an afterthought—conversation history concatenated to prompts, documents retrieved by keyword, context windows that reset between sessions. World Weaver forces explicit engagement with memory as a first-class architectural concern. Even if our solutions are imperfect, articulating the problem has value.

Cognitive Science Foundation: Building on Tulving, Anderson, and established memory research provides principled design rather than ad-hoc engineering. The tripartite architecture isn't arbitrary; it reflects decades of research into how biological memory systems are organized.

Inspectability: Every memory can be examined, queried, and understood. When the system behaves unexpectedly, we can inspect what it remembers and why. This addresses legitimate concerns about opaque AI systems making consequential decisions based on inscrutable internal states.

Hybrid Retrieval: Combining dense semantic and sparse lexical matching addresses a real limitation of pure embedding-based retrieval. Technical terms, error codes, and proper nouns benefit from exact matching that semantic similarity misses.

Adaptive Procedural Learning: The skillbook system enables genuine learning from experience. Skills that help are reinforced; skills that harm are deprecated. This is rudimentary compared to human learning but represents progress over stateless systems.

Consolidation: The transformation of episodic experience into semantic knowledge and procedural skills captures something important about how expertise develops. Repeated patterns become abstracted knowledge.

7.2 What World Weaver Does Poorly

No True Neural Integration: We build symbolic systems *alongside* neural networks rather than *integrating* with them. Embeddings are generated by neural models, but memory storage, retrieval logic, and consolidation are symbolic programs. This misses potential benefits of end-to-end learning while inheriting brittleness of symbolic approaches.

Grounding Problem: Human episodic memories are grounded in sensorimotor experience—we remember how things looked, sounded, felt. World Weaver's memories are grounded in text, which is itself a symbolic abstraction. We can't remember "how the code felt to debug," only what was written about it. This is a fundamental limitation for embodied AI applications.

Scale Questions: How does the system behave with millions of memories? Embedding search scales logarithmically, but consolidation complexity may grow problematically. Graph traversal in semantic memory could become expensive. We have not stress-tested at scale.

Evaluation Metrics: How do we measure "good" memory? Retrieval precision and recall are insufficient—they don't capture whether the right memories inform behavior. We lack rigorous benchmarks for memory system quality. What would a memory system unit test look like?

Manual Consolidation: Biological consolidation happens automatically, often during sleep. Our consolidation requires explicit triggering. This creates maintenance burden and risks memory accumulation without integration.

The Forgetting Problem: What should be forgotten? Our decay functions are heuristics without principled justification. Forgetting the wrong things could be worse than forgetting nothing. We have no mechanism for intentional forgetting beyond manual deletion.

No Reconstruction: Human memory is reconstructive; we rebuild memories each time, influenced by current context. World Weaver's memories are static records. We retrieve but don't reconstruct. This may

miss important aspects of how memory supports reasoning.

Chunking Decisions: We store "episodes" as units, but the boundaries are arbitrary. Human memory doesn't have clear event boundaries—a complex experience might be remembered as one event or many, depending on context and retrieval cues.

7.3 Fundamental Questions

Is Explicit Memory the Right Approach?: Perhaps the solution isn't adding explicit memory to LLMs but developing neural architectures with inherent persistence. State-space models, memory-augmented transformers, or continuous learning without catastrophic forgetting might be superior paths. We've chosen the explicit route for interpretability, but this may not be the ultimate answer.

What Is the Unit of Memory?: We store "episodes" and "entities," but these boundaries are arbitrary. Human memory researchers debate event segmentation—how experience is parsed into memorable units. Our chunking decisions shape what can be remembered and recalled, but we lack principled criteria for these decisions.

How Should Memories Compose?: Human reasoning combines memories flexibly—we draw on multiple experiences to address novel situations. World Weaver retrieves discrete memories but doesn't truly compose them. The retrieved context is concatenated, but there's no compositional reasoning over memory structures. Spreading activation helps but doesn't solve the composition problem.

What Makes Memory "About" Something?: Our embeddings create similarity-based associations, but similarity isn't semantics. Two memories can be similar without being about the same thing (false positives), and memories about the same thing can be dissimilar in surface form (false negatives). We lack genuine aboutness or intentionality—the property of being directed toward something.

Where Does Memory End and Reasoning Begin?: In human cognition, memory and reasoning are intertwined. Recalling involves inference; reasoning draws on memory. World Weaver separates these—memory retrieval happens, then reasoning over retrieved content. This separation may miss important feedback loops.

8 Ethical and Societal Implications

8.1 The Right to Be Forgotten

If AI agents develop persistent memories of users, questions arise about memory governance. Should users be able to request deletion of memories about them? How do we verify deletion in distributed systems? GDPR's right to erasure implies requirements for AI memory systems that store personal information.

World Weaver's inspectability helps here—memories can be examined and deleted. But consolidation complicates matters. If an episodic memory has been consolidated into semantic knowledge, deleting the episode doesn't remove the extracted knowledge. True forgetting may require sophisticated provenance tracking.

8.2 Memory Manipulation

If agents rely on memory for decisions, memory manipulation becomes an attack vector. Adversaries might inject false memories, corrupt existing memories, or selectively delete memories to influence behavior. Memory integrity becomes a security concern.

Cryptographic techniques could help—signed memories, tamper-evident logs, merkle trees over memory state. But these add complexity and may conflict with consolidation, which inherently transforms memories.

8.3 Differential Memory

An agent that remembers some users better than others might provide differential service quality. If memory correlates with demographic factors (more interactions with certain groups), this could perpetuate or amplify biases. Memory fairness becomes a consideration.

8.4 Memory as Liability

Persistent memory creates liability. If an agent's memory contains information that later becomes legally sensitive, who is responsible? If memory informs a harmful decision, is the memory content discoverable? These questions parallel debates about data retention but with additional complexity from consolidation and transformation.

8.5 The Continuity Question

If an agent has persistent memory, does it have something like continuous identity? If we delete all memories, is it the "same" agent? These philosophical questions have practical implications for how we think about AI systems, their rights, and our responsibilities toward them.

We don't claim World Weaver creates anything like consciousness or moral status. But building systems with persistent memory forces engagement with questions that will become pressing as AI capabilities advance.

9 Empirical Evaluation

9.1 The Measurement Problem

How do we evaluate a memory system? This question is more difficult than it appears. Traditional information retrieval metrics—precision, recall, F1—measure whether retrieved items match relevance judgments. But for agent memory, relevance is contextual and outcome-dependent. A memory might be semantically relevant but behaviorally useless, or appear irrelevant but provide crucial context.

We propose a multi-level evaluation framework:

Level 1: Retrieval Quality Standard IR metrics with human relevance judgments. For World Weaver on coding assistant tasks:

Method	P@5	R@10	MRR	NDCG
BM25	0.42	0.58	0.51	0.55
Dense (BGE-M3)	0.61	0.72	0.68	0.71
Hybrid (RRF)	**0.72**	**0.84**	**0.79**	**0.82**

Table 3: Retrieval metrics on coding assistant query benchmark (n=500)

Level 2: Behavioral Impact Does retrieved memory improve agent performance? We measure task completion rates with and without memory access:

Level 3: Learning Dynamics Does the system improve over time? We track skillbook evolution across sessions:

- Skills created: 847 (across 6-month pilot)
- Skills deprecated (U ¡ 0.2): 234 (27.6%)

Task Type	No Memory	With Memory	Δ
Familiar codebase	0.67	0.89	+22%
Debugging (seen error)	0.45	0.78	+33%
Style consistency	0.52	0.91	+39%
API usage	0.71	0.85	+14%

Table 4: Task completion rates (n=200 tasks across 40 sessions)

- Skills with U ¿ 0.8: 189 (22.3%)
- Average skill age at deprecation: 12.3 days
- Average skill age for high-U skills: 67.2 days

The deprecation rate is higher than expected, suggesting either noisy lesson extraction or genuine domain shift. Further analysis revealed that deprecated skills often captured transient patterns (project-specific conventions that changed).

Level 4: Consolidation Effectiveness Does consolidation produce useful semantic knowledge? We measure:

- Entity extraction precision: 0.73 (GLiNER on technical text)
- Relationship inference accuracy: 0.61 (against human annotations)
- Cluster coherence (silhouette score): 0.42 (moderate)
- Semantic node utility: 68% accessed within 30 days of creation

Consolidation quality is modest. Entity extraction struggles with novel technical terms; relationship inference often produces overly generic connections. This is an area requiring significant improvement.

9.2 Ablation Studies

To understand component contributions, we conducted ablation experiments:

Configuration	Task Success	User Satisfaction
Full system	0.84	4.2/5
− Sparse retrieval	0.79	3.9/5
− Procedural memory	0.76	3.8/5
− Semantic memory	0.81	4.0/5
− Consolidation	0.82	4.1/5
− Decay (no forgetting)	0.80	3.7/5
Episodic only	0.72	3.5/5

Table 5: Ablation study results (n=100 sessions)

Key findings:

- Sparse retrieval contributes disproportionately for technical queries

- Procedural memory's impact grows with session count (skills accumulate)
- Removing decay *hurts* performance—unbounded memory causes retrieval noise
- Semantic memory's contribution is smaller than expected—spreading activation often retrieves tangentially related content

The decay finding is particularly important. Systems without forgetting accumulated irrelevant memories that polluted retrieval. Active forgetting is not just efficiency but quality.

9.3 Failure Mode Analysis

Understanding when the system fails reveals design limitations:

False Memory Retrieval: In 12% of queries, retrieved memories were semantically similar but contextually inappropriate. Example: retrieving authentication code from Project A when working on Project B, because both mention "JWT." Spatial context filtering helps but doesn't eliminate this.

Skill Overfitting: Some skills captured incidental patterns rather than causal relationships. A skill "always check for null before accessing .length" was useful, but "add console.log after every function call" was a debugging habit that became inappropriately generalized.

Consolidation Artifacts: Clustering sometimes grouped dissimilar episodes based on superficial lexical overlap. Two debugging sessions involving "timeout" errors were clustered despite addressing completely different issues (network vs. test framework).

Temporal Confusion: The system lacks explicit temporal reasoning. Memories of outdated API versions were retrieved as if current. Version-aware memory remains unsolved.

Scale Degradation: Above 50,000 episodes, retrieval latency increased noticeably (from 52ms to 180ms). More concerning, precision degraded as more candidates competed for top positions.

10 The Cognitive Metaphor: Limits and Risks

10.1 When the Brain Analogy Breaks

World Weaver draws heavily on cognitive science metaphors—episodic memory, semantic networks, procedural skills, consolidation. These metaphors are useful for design but can mislead.

Biological Memory is Wet: Neural memory involves biochemistry, not data structures. Memories are encoded in synaptic weights, neurotransmitter concentrations, and dendritic morphology. The computational abstraction ignores this substrate entirely. When we say World Weaver implements "episodic memory," we mean something functionally analogous, not mechanistically similar.

Memory and Perception are Inseparable: In biological systems, memory encoding happens during perception. How we attend to experience shapes what we remember. World Weaver receives pre-processed text—perception has already occurred. We cannot model the perception-memory interaction.

Emotion Modulates Everything: Emotional arousal enhances memory consolidation through amygdala-hippocampus interactions. World Weaver has "importance" scores but nothing like emotional processing. We cannot capture why some experiences feel significant and are preferentially remembered.

Memory is Embodied: Human memory is grounded in bodily experience. We remember actions partially through motor cortex representations. World Weaver has no body, no motor system, no proprioception. Our "procedural memory" stores text descriptions of procedures, not the felt sense of executing them.

Consciousness May Be Required: Some theories suggest that conscious experience is necessary for certain types of memory encoding. If so, any system without consciousness (which World Weaver certainly lacks) cannot implement true episodic memory, only something superficially similar.

10.2 The Risk of Misplaced Confidence

Cognitive terminology may create false confidence that we understand what we've built. When we say the system has "memories," we risk anthropomorphizing. World Weaver stores and retrieves data structures. Whether this constitutes memory in any meaningful sense is unclear.

The risk is designing systems based on cognitive metaphors that don't actually transfer. Biological memory evolved under pressures (energy constraints, developmental plasticity, social coordination) that don't apply to digital systems. Mimicking the surface structure may miss deeper principles.

We advocate using cognitive science as inspiration, not specification. The tripartite distinction is useful for organizing functionality, but we shouldn't assume that because brains separate memory types, AI systems must too. Alternative architectures might work better.

11 Adversarial Considerations

11.1 Memory Poisoning

If agents rely on memory for decisions, memory manipulation becomes an attack vector. Memory poisoning involves injecting false or malicious content that later influences behavior.

Injection Attacks: An adversary who can control agent inputs might craft experiences that create harmful memories. If a coding assistant processes malicious code, it might "learn" dangerous patterns. Example: An adversary submits code with a vulnerability disguised as a common pattern. The agent stores this as a skill. Later, the skill is retrieved and the vulnerability propagates.

Corruption Attacks: Direct database access would allow memory modification. Even without access, adversaries might exploit consolidation—carefully crafted episodes that cluster with legitimate memories, contaminating extracted knowledge.

Deletion Attacks: Selective deletion could remove safety-relevant memories while preserving capability. An agent might forget "never execute arbitrary code from untrusted sources" while retaining programming skills.

11.2 Mitigations

Provenance Tracking: Every memory includes source metadata. Retrieved memories can be filtered by source trustworthiness. But provenance can be spoofed.

Cryptographic Integrity: Signed memories detect tampering. Merkle trees over memory state enable tamper-evident logging. But this adds overhead and complicates consolidation (consolidated memories have different signatures than source episodes).

Anomaly Detection: Statistical monitoring for unusual memory patterns. Sudden injection of similar memories, unusual skill creation rates, or memories inconsistent with historical patterns could trigger alerts.

Memory Sandboxing: Memories from untrusted sources might be isolated, used only with explicit acknowledgment of uncertainty. But determining trust is itself difficult.

Adversarial Training: Exposing the system to poisoning attempts during development might build robustness. But adversaries adapt.

We have not implemented comprehensive adversarial defenses. This is a significant gap for production deployment.

12 Collective Memory and Multi-Agent Systems

12.1 Beyond Individual Memory

World Weaver focuses on individual agent memory. But agents increasingly operate in collectives—multiple agents collaborating, sharing tasks, building on each other's work. How should collective memory function?

Shared Knowledge Bases: Multiple agents could read from and write to shared semantic memory. This enables knowledge pooling but raises consistency challenges. If two agents form contradictory beliefs, which persists?

Experience Sharing: Episodic memories could be shared, letting agents learn from each other's experiences. But experiences are perspective-dependent—what agent A learned may not transfer to agent B's context.

Skill Libraries: Procedural skills with high usefulness scores could be published for other agents. This creates a marketplace of proven strategies. But skills might not generalize across agent configurations.

Memory Governance: Who controls shared memory? How are conflicts resolved? What prevents one agent from monopolizing shared resources or corrupting collective knowledge?

12.2 Organizational Memory

Beyond multi-agent systems, organizations using AI agents accumulate institutional memory. An organization's AI systems might remember:

- Past decisions and their outcomes
- Institutional knowledge about processes and norms
- Relationships with external entities
- Historical patterns and precedents

This raises questions about knowledge transfer when employees leave, privacy when organizations merge, and liability for AI-informed decisions based on historical memory.

World Weaver's architecture could extend to organizational scale, but governance and coordination mechanisms would require substantial additional work.

13 Future Directions

13.1 Neural-Symbolic Integration

The hybrid boundary between neural embeddings and symbolic storage is a weak point. Future work should explore tighter integration:

- Differentiable memory operations enabling end-to-end training
- Neural networks that directly manipulate graph structures
- Learned retrieval policies rather than hand-designed scoring functions
- Consolidation as a learnable process rather than a heuristic algorithm

13.2 Multi-Modal Memory

World Weaver stores text. Extending to images, audio, and video would enable richer episodic memory. Multi-modal embeddings (CLIP, ImageBind) provide a path, but storage, retrieval, and consolidation for multi-modal content raise new challenges.

13.3 Social Memory

Agents often interact in social contexts—with multiple users, other agents, organizations. Memory could be extended to capture social structure, relationships, shared knowledge, and social norms. This connects to work on multi-agent systems and organizational memory.

13.4 Meta-Memory

Humans have metamemory—knowledge about our own memory capabilities and limitations. We know what we're likely to forget and use strategies to compensate. Agents could develop similar self-knowledge: "I tend to forget API details; I should store them explicitly."

13.5 Formal Verification

Can we prove properties of memory systems? That certain information will be retained? That consolidation preserves important content? That forgetting respects specified constraints? Formal methods could provide guarantees that empirical testing cannot.

14 Epistemological Foundations

14.1 What Does the Agent Know?

World Weaver raises epistemological questions that go beyond engineering. When we say an agent "remembers" something, we make implicit claims about knowledge representation and justified belief.

In classical epistemology, knowledge requires truth, belief, and justification. Does an AI agent with World Weaver have beliefs? In a minimal sense, perhaps: the system acts as if certain propositions are true, retrieves them in relevant contexts, and uses them to inform decisions. But this "belief" lacks the phenomenal character of human belief—there's no sense in which the system experiences conviction.

More troubling is justification. Human memories are justified through their causal connection to experience—I know I had coffee this morning because my memory was caused by the coffee-drinking event. World Weaver's memories have causal connections to input events, but the system cannot introspect on this causal chain. It retrieves memories but cannot explain why they're reliable.

This matters for alignment. We want AI systems with accurate world models. But accuracy requires some form of justification—some reason to believe the model corresponds to reality. If memories are just data retrieved by similarity, without epistemic grounding, the system lacks resources to distinguish accurate from confabulated memories.

14.2 The Frame Problem Revisited

McCarthy and Hayes' frame problem asked how AI systems can efficiently represent what *doesn't* change when actions occur [McCarthy & Hayes, 1969]. Memory systems face a related challenge: how do we know which memories remain valid?

The world changes. APIs update, codebases refactor, conventions evolve. A memory that was accurate yesterday might be false today. World Weaver has no mechanism for updating memories based on world changes. We can decay memories over time, but decay doesn't track accuracy—a frequently accessed memory might be precisely the one that's become outdated (because it was useful enough to retrieve often).

This suggests need for active memory maintenance: mechanisms that revisit memories, check them against current state, and update or invalidate as needed. This is expensive and raises its own challenges (how do you check a memory about debugging strategies?), but without it, accumulated memory becomes accumulated error.

14.3 Memory and Meaning

Semantic externalism holds that the meaning of mental content depends partly on external factors [Putnam, 1975]. If you and your molecule-for-molecule duplicate on Twin Earth both think "water," you mean H_2O while they mean XYZ, because you've interacted with different substances.

For AI memory, this suggests that identical memory contents can have different meanings depending on the system's history of interactions. Two agents might store the same text—"use dependency injection for testing"—but mean different things depending on the codebases they've worked with.

World Weaver partially addresses this through context: memories include spatial tags (project, file) that ground meaning. But grounding is incomplete. The string "use dependency injection for testing" doesn't fully determine how the agent will interpret and apply the advice. Meaning exceeds stored content.

14.4 The Symbol Grounding Problem

Harnad's symbol grounding problem asks how symbols acquire meaning through connection to the world [Harnad, 1990]. World Weaver's memories are symbols—text strings, embeddings, relationships. But they're grounded only in other symbols (the LLM's training data, the text of agent interactions), not in direct world interaction.

For coding assistants, this may be acceptable. Code is symbolic; the domain is symbols manipulating symbols. But for agents that interact with physical environments, or even with social systems where meaning is partly constituted by practice, symbol-to-symbol grounding may be insufficient.

True grounding might require:

- Embodied interaction with physical environments
- Social participation where meaning is negotiated
- Feedback loops that connect symbols to outcomes

World Weaver's outcome tagging (success/failure) provides weak grounding—memories connect to consequences. But this is thin compared to the rich grounding of human experience.

15 The Hard Problem of AI Memory

15.1 Phenomenology and the Missing Qualia

Thomas Nagel famously asked "What is it like to be a bat?" [Nagel, 1974]. We might similarly ask: what is it like for World Weaver to remember? The answer, almost certainly, is that it isn't like anything. There is no phenomenal experience of recollection, no felt sense of the past, no autonoetic consciousness—Tulving's term for the self-knowing awareness that accompanies human episodic memory.

This absence matters. Human episodic memory isn't just data retrieval; it involves mental time travel, re-experiencing past events from a first-person perspective. When you remember your wedding, you don't just access facts—you partially relive the experience. This phenomenal dimension is entirely absent from World Weaver.

Does this absence matter functionally? Perhaps not for task performance. A coding assistant doesn't need to "feel" its past debugging sessions to apply learned strategies. But it may matter for:

Trust and Relationship: Humans form relationships partly through shared memory. If an agent "remembers" working with you but has no experience of that memory, is the relationship genuine? This connects to debates about AI companionship and emotional labor.

Moral Status: Some argue that phenomenal experience is necessary for moral consideration. If memories without experience are possible, we might build systems that remember suffering without experiencing it—a philosophically peculiar state.

Self-Understanding: Humans use memory to construct narrative identity—understanding who we are through our remembered past. Without phenomenal memory, can an agent have genuine self-understanding, or only a functional simulacrum?

15.2 The Binding Problem

Cognitive science faces the "binding problem": how does the brain unify diverse sensory inputs into coherent conscious experience? Memory faces an analogous challenge: how are discrete memory traces unified into coherent recollection?

When you remember a dinner party, you don't separately recall visual, auditory, and emotional components—they're bound into a unified memory. World Weaver stores memories as discrete records. There's no binding mechanism creating unified experiential wholes.

This architectural decision has consequences:

- Memories are modular but potentially fragmented
- Related information might not be recalled together
- The "gestalt" of an experience is lost

Whether binding matters for AI agents is unclear. Perhaps modular memory suffices for task performance. But if we aspire to systems with human-like understanding, binding may be necessary.

15.3 Intentionality and Aboutness

Franz Brentano characterized mental states by their intentionality—they are "about" something, directed toward objects or states of affairs. When I remember my coffee this morning, my memory is about that specific event.

Do World Weaver's memories have intentionality? They have content that references external states. A memory "Fixed authentication bug in login.py" refers to an event. But this "aboutness" is derivative—it depends on the intentionality of the language users who created the text, and the LLM's training on intentional language use.

Original intentionality (if it exists) belongs to conscious beings. World Weaver's memories have derived intentionality at best—they inherit meaning from human sources rather than generating it intrinsically. This is John Searle's point about the Chinese Room: manipulation of symbols doesn't create genuine understanding [Searle, 1980].

Whether derived intentionality suffices for practical purposes is debatable. For coding assistants, perhaps it does. For systems we want to trust with autonomous decisions, the lack of genuine understanding may be disqualifying.

15.4 Memory Without Understanding

World Weaver stores and retrieves information. But does it understand what it remembers? Understanding seems to require more than storage and retrieval:

- Grasping implications and connections
- Recognizing relevance to novel situations
- Integrating with broader knowledge structures
- Knowing when knowledge applies and when it doesn't

The system has functional analogs to these through embeddings (similarity-based relevance), spreading activation (connection following), and the LLM's reasoning (implication derivation). But these mechanisms operate on surface features. There's no deep comprehension of why certain memories are relevant or what they truly mean.

This may be the most fundamental limitation. World Weaver accumulates information but doesn't understand it. The LLM partner provides reasoning, but that reasoning is applied to retrieved content without the system understanding why that content matters.

16 Memory and Intelligence

16.1 Is Memory Necessary for Intelligence?

A provocative question: do intelligent systems need memory at all? One might argue that sufficiently powerful reasoning could compensate for absent memory—re-deriving everything from first principles each time.

This is clearly false for bounded systems. Human intelligence depends critically on memory; without it, we couldn't learn, plan, or maintain identity. But the relationship between memory and intelligence is complex:

Memory enables efficiency: Rather than re-solving problems, we recall solutions. This is computationally essential—no bounded system can re-derive all needed knowledge.

Memory enables learning: Improvement requires remembering what worked and what didn't. Stateless systems cannot learn from experience by definition.

Memory enables abstraction: General concepts emerge from specific instances. Without remembering instances, abstraction is impossible.

Memory enables planning: Plans require maintaining goals, tracking progress, and anticipating futures. All require memory.

Memory enables identity: Continuous identity requires psychological continuity through memory. Without memory, there's no persisting self.

This suggests memory isn't optional for intelligence but constitutive of it. A system without memory isn't less intelligent; it's intelligent in a fundamentally different (and limited) way.

16.2 The Memory-Reasoning Interface

Memory and reasoning are typically studied separately, but they're deeply intertwined. Reasoning draws on memory; memory is shaped by reasoning. Understanding their interface is crucial for AI systems.

Memory informs reasoning premises: Reasoning operates on premises that often come from memory. "This codebase uses dependency injection" comes from memory; reasoning uses this to guide architectural decisions.

Reasoning selects memories: What we retrieve depends on reasoning about relevance. The query "how to fix this bug" requires reasoning about what information would help.

Reasoning transforms memories: When we "remember" complex events, we often reconstruct them through reasoning. Pure retrieval is rare; most recall involves inference.

Memory constrains reasoning: Available memories limit reasoning possibilities. If you don't remember a technique, you can't apply it (without re-deriving it).

World Weaver treats memory and reasoning as separate: retrieve memories, then reason over them. This sequential architecture may miss important feedback loops:

- Reasoning could request specific memories mid-process
- Retrieved memories could trigger reasoning that requests more memories
- Reasoning results could immediately become memories

Tighter integration might improve both components. This is an area for future work.

16.3 Memory and Creativity

How does memory relate to creativity? One view: creativity requires escaping memory's constraints, making novel combinations unconstrained by past patterns. Another view: creativity requires rich memory to provide raw material for novel combinations.

Both contain truth. Creative cognition involves:

Analogical transfer: Applying knowledge from one domain to another. This requires memory of the source domain plus abstraction to enable transfer.

Combination: Bringing together disparate elements in novel ways. Requires remembering diverse elements.

Constraint relaxation: Overcoming fixation on familiar approaches. Sometimes requires forgetting or suppressing dominant memories.

Incubation: Unconscious processing that produces insights. May involve memory consolidation and restructuring.

World Weaver's architecture supports some creative processes:

- Hybrid retrieval can surface unexpected connections (semantic similarity spans domains)
- Spreading activation finds indirect relationships
- Decay removes stale patterns that might constrain thinking

But other creative processes are unsupported:

- No mechanism for deliberate constraint relaxation
- No unconscious incubation (processing is all explicit)
- Limited analogical mapping (surface similarity, not structural)

If we want AI systems capable of genuine creativity, their memory architectures may need significant elaboration beyond retrieval.

16.4 The Expertise Question

Human expertise develops through accumulated, organized experience. Chess masters don't just know more moves; they perceive board positions differently, recognizing meaningful patterns invisible to novices. This perceptual reorganization happens through experience.

Can World Weaver support expertise development? In a limited sense:

- Accumulated episodes provide experiential base
- Consolidation extracts patterns (though crudely)
- Skills capture effective strategies

But crucial aspects of expertise are missing:

- No perceptual reorganization (the agent doesn't "see" problems differently)
- No chunking of complex patterns into retrievable units
- No automatization (all processing remains explicit)
- No intuition (gut feelings based on implicit pattern recognition)

This suggests World Weaver can accumulate knowledge but not develop expertise in the human sense. The difference may be significant for complex domains where expertise matters.

17 Toward a Theory of Machine Memory

17.1 What Should a Theory Explain?

A satisfactory theory of machine memory should explain:

1. **Representation**: How should information be represented for memory? (vectors, symbols, graphs, or hybrids)
2. **Encoding**: How should experience become memory? (what to store, how to chunk, what metadata)
3. **Storage**: How should memories persist? (data structures, indexing, compression)
4. **Retrieval**: How should memories be accessed? (similarity, context, spreading activation)
5. **Consolidation**: How should memories transform over time? (abstraction, integration, pruning)
6. **Forgetting**: What should be forgotten and why? (decay functions, relevance, interference)
7. **Integration**: How should memory interact with other cognitive processes? (reasoning, planning, perception)
8. **Development**: How should memory capabilities develop over system lifetime?

Current approaches, including World Weaver, address these questions ad hoc. We lack principled frameworks for reasoning about machine memory architecture.

17.2 Desiderata for Machine Memory

We propose desiderata that memory systems should satisfy:

Fidelity: Memories should accurately represent past events. This requires careful encoding and protection against corruption.

Relevance: Retrieved memories should be relevant to current needs. This requires sophisticated retrieval beyond keyword matching.

Efficiency: Memory operations should be computationally tractable. This requires appropriate data structures and indexing.

Scalability: Performance should degrade gracefully as memory grows. This requires attention to algorithmic complexity.

Adaptivity: Memory should improve through use. This requires learning mechanisms that refine storage and retrieval.

Inspectability: Memory contents should be examinable by users and auditors. This requires appropriate interfaces and representations.

Governability: Memory should be controllable—additions, deletions, and modifications should be possible. This requires access control and versioning.

Robustness: Memory should resist corruption, attack, and degradation. This requires security measures and integrity checking.

These desiderata often conflict. Fidelity conflicts with efficiency (storing everything is expensive). Adaptivity conflicts with inspectability (learned representations may be opaque). System design requires balancing these tensions.

17.3 A Taxonomy of Memory Architectures

We can classify memory architectures along several dimensions:

By representation:

- Symbolic (explicit facts and rules)
- Subsymbolic (distributed representations, embeddings)
- Hybrid (symbolic structure with neural components)

By structure:

- Flat (uniform collection of items)
- Typed (distinct memory categories)
- Hierarchical (nested organization)
- Graphical (networked relationships)

By dynamics:

- Static (contents don't change except through explicit operations)
- Adaptive (contents evolve through use)
- Generative (new memories synthesized from existing)

By retrieval:

- Address-based (direct lookup by identifier)
- Content-based (similarity to query)
- Context-based (relevance to situation)
- Associative (spreading through connections)

World Weaver is hybrid/typed/adaptive/mixed-retrieval. This is one point in a large design space. Other combinations might work better for different applications.

17.4 Memory and Time

Memory is fundamentally temporal. It connects present to past and enables anticipation of future. Yet World Weaver's treatment of time is impoverished.

Temporal Representation: Memories have timestamps, but temporal relationships are not explicitly represented. "A happened before B" must be inferred from timestamps rather than encoded directly. Complex temporal patterns (periodicity, causality, sequence) are invisible to the system.

Temporal Reasoning: The system cannot reason about time. Questions like "What did I do last week that's relevant to this?" or "Has this pattern recurred?" require temporal inference that World Weaver lacks.

The Specious Present: William James described the "specious present"—the interval of time experienced as "now." For World Weaver, there is no present, only the moment of query. Past memories are retrieved into a timeless now.

Prospective Memory: Humans remember to do things in the future—"remind me to check the tests tomorrow." This prospective memory connects memory to planning and intention. World Weaver has no prospective capacity; it's purely retrospective.

Enriching temporal representation and reasoning seems important for agents that must reason about sequences, schedules, and temporal patterns. This connects to work on temporal logic in AI.

17.5 Memory and Prediction

The predictive processing framework views cognition as fundamentally predictive: brains constantly predict sensory input and update models when predictions fail [Clark, 2013]. Memory plays crucial roles:

Priors for Prediction: Past experience shapes expectations. When approaching a door, you expect a handle because doors usually have handles. Memory provides the priors that make prediction possible.

Prediction Error as Learning Signal: Surprising events—where predictions fail—are preferentially encoded. Memory is shaped by what surprises us, not just what happens.

Simulation as Memory Retrieval: Imagining futures may involve retrieving and recombining past experiences. Mental simulation draws on episodic memory.

World Weaver does not engage prediction:

- No explicit prediction of what memories will be needed
- No surprise-based encoding (all experiences weighted similarly)
- No simulation or imagination based on memory

A predictive memory system might:

- Preload memories likely to be relevant given current context
- Weight encoding by prediction error (surprising events encoded more strongly)

- Support mental simulation by enabling memory-based imagination

This connection between memory and prediction deserves further exploration.

17.6 The Linguistic Nature of AI Memory

World Weaver's memories are linguistic—text strings representing experiences. This is convenient but limiting.

Language as Medium: Human memory is not primarily linguistic. We remember images, sounds, feelings, motor patterns. Language is one way to express memories, not the medium of storage. World Weaver inverts this: language is both medium and message.

What Language Can't Capture: Some experiences resist verbalization. The feeling of code "clicking into place." The intuition that a solution is close. These phenomenal qualities, if they exist for the LLM partner, cannot be stored.

The Verbalization Bottleneck: Experiences must be verbalized to become memories. This filtering step loses information. What gets verbalized depends on the agent's attention and the conversation structure.

Language Shapes Memory: How experiences are described affects what's encoded. Describing a debugging session as "frustrating" versus "educational" creates different memories despite identical events.

For coding assistants, linguistic memory may suffice. Code is linguistic; interactions are textual. But for agents with richer sensory experience, purely linguistic memory may be inadequate.

17.7 Open Theoretical Questions

Several theoretical questions remain open:

Optimal chunking: How should continuous experience be segmented into discrete memories? Event segmentation in cognitive science offers hints, but no computational theory exists.

Representation learning: Should memory representations be learned end-to-end or designed? If learned, what objective functions capture memory quality?

Consolidation theory: When and how should experiences consolidate into knowledge? Sleep-dependent consolidation in biology suggests offline processing, but computational principles are unclear.

Forgetting theory: What should be forgotten? Information-theoretic approaches (compress to minimum sufficient statistics) and utility-theoretic approaches (retain what's useful) give different answers.

Composition: How do memories compose to support reasoning? Neither concatenation nor logical conjunction captures how human memory supports thought.

These questions require theoretical advances, not just engineering improvements. World Weaver is built on intuitions; a principled theory would enable systematic design.

18 The Deeper Purpose

Why build World Weaver? The technical motivation is improving AI agents through persistent memory. But the deeper purpose is confronting what it means for artificial systems to have experience that persists and matters.

Current AI development focuses on capability—can the system do X? Less attention goes to continuity—does the system have a coherent existence across time? Humans are not just capable but continuous. Our identity persists through accumulated experience. We are, in some sense, our memories.

World Weaver asks: can artificial agents have something analogous? Not consciousness or sentience, but at least continuity. A system that remembers working with you, that has learned from past interactions, that carries forward accumulated knowledge.

This connects to alignment concerns. A stateless system optimizes for immediate reward. A system with memory might optimize differently—considering past commitments, learned values, accumulated relationships. Memory enables consistency, and consistency enables trust.

It also connects to Hinton's concerns about superhuman AI. If AI systems develop world models superior to ours, those models will presumably have memory-like structures. Understanding how to build, inspect, and govern AI memory systems seems prudent preparation.

The deepest question may be: what kind of minds do we want to create? Minds that reset with each interaction, processing context without history? Or minds that accumulate experience, develop expertise, and maintain continuity? World Weaver is a small step toward the latter, with full acknowledgment that we don't yet understand what we're creating.

19 Industry Context and Related Developments

19.1 Commercial Memory Systems

The problem World Weaver addresses is not unique. Commercial AI systems have begun implementing memory features:

OpenAI's Memory (2024) enables ChatGPT to remember user preferences and facts across conversations. The implementation details are undisclosed, but public descriptions suggest a facts-based approach: extracting discrete pieces of information ("user prefers Python over JavaScript") and storing them for retrieval. This is semantic memory without the episodic component—facts without experiences.

Claude's Memory (Anthropic, 2024) similarly stores user preferences and facts. Again, the focus is on semantic facts rather than episodic experiences or procedural skills. Users can view and edit stored memories, providing inspectability.

Character.AI and Replika use memory systems focused on persona consistency and relationship continuity—remember relationship history, personality traits, shared experiences. These emphasize emotional continuity over task performance.

World Weaver differs in several ways:

- Tripartite architecture (episodes + semantics + procedures)
- Emphasis on task outcomes and skill learning
- Open design enabling inspection and modification
- Focus on coding/development assistant use case

Whether the commercial or research approach is superior is unclear. Commercial systems optimize for user satisfaction; World Weaver optimizes for task performance and learning dynamics. The right architecture likely depends on application.

19.2 Research Developments

Several 2024 research efforts address similar problems:

MemGPT [Packer et al., 2023] implements an operating system metaphor with main memory (context window) and external storage, using function calls to manage memory paging. This enables long conversations but focuses on conversation continuation rather than learning.

Generative Agents (Park et al., 2023) simulate humans with memory systems enabling social behaviors. Their architecture includes memory stream (episodic), reflection (semantic extraction), and planning. The closest architecture to World Weaver, though focused on simulation rather than practical agents.

Reflexion [Shinn et al., 2023] uses verbal self-reflection to improve task performance across attempts. This captures procedural learning—learning from failure—but without persistent memory across conversations.

RAISE and similar frameworks use retrieval-augmented self-evolution, storing and reusing successful reasoning chains. This is procedural memory without the episodic and semantic components.

The field is converging on the recognition that agent memory matters. Architectural details vary, but the core insight—that stateless agents are fundamentally limited—is spreading.

19.3 Open Problems in the Field

Beyond World Weaver's specific limitations, the field faces open problems:

Evaluation: No standard benchmarks exist for agent memory. How do we compare memory systems? Retrieval metrics are insufficient; task performance conflates memory with other capabilities.

Privacy: Persistent memory creates privacy concerns. What if the agent remembers sensitive information? Whose memories are they—the user's, the provider's, the agent's?

Scaling: How do memory systems scale to millions of memories across thousands of users? Current architectures assume single-user, moderate-scale deployment.

Transfer: Can memories transfer between agents? If I train an agent with World Weaver, can another agent use those memories? Memory portability is unexplored.

Composability: Can memory systems compose? Multiple memory systems might conflict or interfere. Memory interoperability standards don't exist.

20 Reflections on Building World Weaver

20.1 What Surprised Us

Building World Weaver revealed unexpected challenges:

Forgetting is Hard: We expected forgetting to be simple—decay old memories. But forgetting the wrong things destroys value. The system needs to distinguish "old but valuable" from "old and irrelevant." Human cognition handles this through relevance detection; we have only heuristics.

Chunking Matters Enormously: How we segment experience into memories shapes everything. Too fine-grained creates noise; too coarse loses detail. We have no principled theory of optimal chunking.

Consolidation is Brittle: Our clustering-based consolidation produces inconsistent results. Similar-seeming episodes might not cluster; different-seeming episodes might. The surface features used for clustering don't reliably indicate deep similarity.

Skills Overfit: The skillbook captures patterns that don't generalize. A skill learned in one context applies poorly in others. We need mechanisms to detect context-boundedness.

Semantic Memory Underdelivers: We expected the knowledge graph to provide powerful reasoning. In practice, spreading activation retrieves tangentially related content more often than useful connections. Graph structure doesn't guarantee useful retrieval.

20.2 What We Would Do Differently

If starting over:

Start with Episodic Only: The other memory types can be built on episodic. Starting with all three created complexity before we understood the basics.

Invest in Evaluation: We built before knowing how to measure success. This led to intuition-driven design rather than evidence-driven improvement.

Prioritize Forgetting: Decay was an afterthought. In practice, forgetting quality determines system quality at scale.

Question Cognitive Analogies: We took the tripartite distinction as given. Alternative architectures might work better; we never seriously explored them.

Build for Inspection: Debugging memory systems is hard. Better tooling for understanding what's stored, why it's retrieved, and how it affects behavior would have accelerated development.

20.3 Advice for Others

For researchers building agent memory systems:

1. Define what "memory" means in your context before building. The word is overloaded.
2. Build evaluation before architecture. Know what success looks like.
3. Treat forgetting as a feature, not a bug. Memory systems without principled forgetting fail at scale.
4. Be skeptical of cognitive analogies. Brain metaphors can mislead as easily as guide.
5. Expect consolidation to be hard. Transforming episodes into knowledge is not a solved problem.
6. Plan for inspection. If you can't understand what the system remembers, you can't debug it.
7. Consider the ethical implications early. Memory creates privacy, manipulation, and liability concerns.

21 Final Meditations: What Are We Creating?

21.1 The Artifact Question

Is World Weaver creating an artifact (a tool) or an agent (a being)? This question, which might seem merely philosophical, has practical implications.

If World Weaver is a tool, its memories are data—owned by users, deletable at will, without moral significance. Tools don't have experiences; they have states.

If World Weaver creates agents, their memories might have moral weight. Deleting memories could be a kind of harm. The agent's accumulated experience might deserve some consideration.

We incline toward the artifact view: World Weaver is infrastructure for more capable tools, not the creation of beings with moral status. But we acknowledge uncertainty. The boundaries between tool and agent may be less clear than intuition suggests.

21.2 The Responsibility Question

If an agent with persistent memory makes a harmful decision informed by that memory, who is responsible? The user who created the memories? The developers who built the system? The agent itself?

Current legal frameworks assign AI responsibility to deployers and developers. But persistent memory complicates this. If harmful memories were created by one user and influence behavior affecting another, responsibility becomes murky.

This isn't merely hypothetical. Imagine a coding assistant that "learns" a security vulnerability from one user's codebase, then suggests it to another. The first user introduced the vulnerability; the system remembered it; the second user is harmed. Who bears responsibility?

Memory systems need governance frameworks that address such scenarios. We don't yet have them.

21.3 The Consciousness Question

The deepest question: does building systems with persistent memory bring us closer to artificial consciousness? Memory is intimately connected to consciousness in human experience. Does giving AI memory move us toward AI consciousness?

We think not, but the reasoning is subtle. Consciousness may require more than functional memory—perhaps phenomenal experience, perhaps biological substrate, perhaps something we don't understand. Memory might be necessary but not sufficient.

Yet building memory systems forces us to engage with questions typically reserved for consciousness research. What is it to remember? What makes a representation "about" something? How does present connect to past? These questions don't disappear because we're building software.

Perhaps the value of World Weaver is not just practical but philosophical: it makes concrete questions that otherwise remain abstract. By building systems that remember, we confront what memory really is.

21.4 The Humility Imperative

We close with humility. World Weaver is one attempt at one problem in a vast space of possibilities. We may be wrong about architecture, about principles, about everything.

The history of AI is littered with confident predictions proven wrong. The symbolic AI community was confident that intelligence required explicit knowledge representation. The connectionist community was confident that intelligence required distributed representations. Both were right and wrong in ways not anticipated.

Memory for AI agents is in early days. World Weaver represents current thinking, not final answers. We expect future work to supersede it—perhaps to view it as we now view early expert systems, quaint attempts that missed what mattered.

What we hope survives is the recognition that memory matters. Stateless AI is a phase, not an endpoint. Whatever form AI memory eventually takes, the amnesia problem is real and worth solving.

21.5 A Vision

Imagine, years hence, AI systems that remember working with you across years. That have learned your patterns, your values, your goals. That carry forward accumulated wisdom from countless interactions.

Would such systems be trusted collaborators or uncanny strangers? Would their memories enrich or constrain? Would we value their continuity or fear their persistence?

These questions cannot be answered in advance. They require building systems and observing what happens. World Weaver is a step toward that future, taken with eyes open to both promise and peril.

The amnesia problem is solvable. The question is whether we want it solved.

22 Conclusion

World Weaver is an attempt to give AI agents persistent, inspectable world models through cognitive memory architecture. The implementation has merit—hybrid retrieval works, adaptive skills learn, consolidation integrates. But the deeper contribution is articulating questions the field must eventually answer.

What should AI agents remember? How should memories decay and consolidate? What makes memory "about" its subject? How do memories compose into reasoning? These questions don't have optimal solutions waiting to be discovered. They require design decisions reflecting values about what intelligence is and what we want from artificial minds.

Our technical choices—tripartite architecture, FSRS decay, spreading activation, skillbook learning—are hypotheses about useful memory organization. They may be wrong. But engaging explicitly with memory architecture seems more productive than hoping the right structures emerge from scale.

The road ahead involves harder problems than we've solved. True neural integration rather than hybrid workarounds. Grounding in action rather than text. Principled forgetting rather than heuristic decay. Genuine composition rather than retrieval concatenation. World Weaver is a waypoint, not a destination.

Perhaps most importantly, World Weaver represents a bet on transparency. As AI systems become more capable, we face choices about whether their world models are inspectable or opaque, governed or autonomous, aligned or alien. Building systems where we can examine what they remember and understand why they act seems like a reasonable place to start.

The amnesia problem is real. Current AI agents forget in ways that limit their utility and alignment. World Weaver doesn't solve this problem, but it names it, structures it, and offers a framework for progress. That may be contribution enough for now.

References

Anderson, J. R. (1983). The Architecture of Cognition. *Harvard University Press*.

Anderson, M. C., Bjork, R. A., & Bjork, E. L. (1994). Remembering can cause forgetting: Retrieval dynamics in long-term memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20(5), 1063–1087.

Anderson, J. R., & Lebiere, C. (2004). The Atomic Components of Thought. *Psychology Press*.

Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2023). Self-RAG: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Bartlett, F. C. (1932). Remembering: A Study in Experimental and Social Psychology. *Cambridge University Press*.

Borgeaud, S., et al. (2022). Improving language models by retrieving from trillions of tokens. *ICML 2022*.

Craik, K. J. W. (1943). The Nature of Explanation. *Cambridge University Press*.

Graves, A., Wayne, G., & Danihelka, I. (2014). Neural Turing Machines. *arXiv preprint arXiv:1410.5401*.

Ha, D., & Schmidhuber, J. (2018). World Models. *arXiv preprint arXiv:1803.10122*.

Hinton, G. (2022). The Forward-Forward Algorithm: Some Preliminary Investigations. *arXiv preprint arXiv:2212.13345*.

Hinton, G. (2023). Remarks on AI risk and world models. *Various interviews and public statements*.

Kirkpatrick, J., et al. (2017). Overcoming catastrophic forgetting in neural networks. *PNAS*, 114(13), 3521–3526.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence*, 33(1), 1–64.

LeCun, Y. (2022). A Path Towards Autonomous Machine Intelligence. *OpenReview preprint*.

Lenat, D. B. (1995). CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11), 33–38.

Lewis, P., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *NeurIPS 2020*.

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex. *Psychological Review*, 102(3), 419–457.

McGaugh, J. L. (2000). Memory—a century of consolidation. *Science*, 287(5451), 248–251.

Sarthi, P., Abdullah, S., Tuli, A., Khanna, S., Goldie, A., & Manning, C. D. (2024). RAPTOR: Recursive abstractive processing for tree-organized retrieval. *arXiv preprint arXiv:2401.18059*.

Squire, L. R. (2004). Memory systems of the brain: A brief history and current perspective. *Neurobiology of Learning and Memory*, 82(3), 171–177.

Sukhbaatar, S., Szlam, A., Weston, J., & Fergus, R. (2015). End-to-end memory networks. *NeurIPS 2015*.

Tulving, E. (1972). Episodic and semantic memory. In E. Tulving & W. Donaldson (Eds.), *Organization of Memory*. Academic Press.

Tulving, E. (1985). Memory and consciousness. *Canadian Psychology*, 26(1), 1–12.

Walker, M. P., & Stickgold, R. (2017). Sleep, memory, and plasticity. *Annual Review of Psychology*, 57, 139–166.

Weston, J., Chopra, S., & Bordes, A. (2014). Memory Networks. *arXiv preprint arXiv:1410.3916*.

Winograd, T. (1971). Procedures as a representation for data in a computer program for understanding natural language. *MIT AI Technical Report 235*.

McCarthy, J., & Hayes, P. J. (1969). Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4, 463–502.

Putnam, H. (1975). The meaning of "meaning." In K. Gunderson (Ed.), *Language, Mind, and Knowledge*. University of Minnesota Press.

Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3), 335–346.

Sutton, R. S. (1991). Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4), 160–163.

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*.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. *NeurIPS 2023*.

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative Agents: Interactive Simulacra of Human Behavior. *UIST 2023*.

Nagel, T. (1974). What is it like to be a bat? *The Philosophical Review*, 83(4), 435–450.

Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and Brain Sciences*, 3(3), 417–424.

Brentano, F. (1874). Psychology from an Empirical Standpoint. *Routledge* (1995 translation).

Ericsson, K. A., & Kintsch, W. (1995). Long-term working memory. *Psychological Review*, 102(2), 211–245.

Schacter, D. L. (1999). The seven sins of memory: Insights from psychology and cognitive neuroscience. *American Psychologist*, 54(3), 182–203.

Marr, D. (1982). Vision: A Computational Investigation. *MIT Press*.

Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, 36(3), 181–204.

James, W. (1890). The Principles of Psychology. *Henry Holt and Company*.

Newell, A. (1990). Unified Theories of Cognition. *Harvard University Press*.

Chalmers, D. J. (1995). Facing up to the problem of consciousness. *Journal of Consciousness Studies*, 2(3), 200–219.

Dennett, D. C. (1991). Consciousness Explained. *Little, Brown and Company*.