World Weaver: Cognitive Memory Architecture for Persistent World Models in Agentic AI Systems

Aaron W. Storey, *Member, IEEE*

Abstract*—Large language models reason and generate code with increasing skill, yet they remain entirely stateless—each interaction begins without memory of past sessions, accumulated knowledge, or learned skills. This paper presents World Weaver, a tripartite cognitive memory architecture that gives AI agents persistent, inspectable world models. We situate this work within Geoffrey Hinton's theory of world models and Yann LeCun's vision of autonomous machine intelligence. The architecture implements episodic, semantic, and procedural memory stores following established cognitive science principles from Tulving's memory taxonomy and Anderson's ACT-R framework. Our hybrid retrieval mechanism combines dense semantic vectors with sparse lexical matching via Reciprocal Rank Fusion, achieving 84% Recall@10 (±0.02) versus 72% for dense-only search ($p < 0.001$). We present implementation details using BGE-M3 embeddings and GLiNER entity extraction, along with ablation studies demonstrating that active forgetting improves retrieval quality. Through systematic survey of 52 papers on AI agent memory (2020–2024) and critical analysis informed by six specialized expert reviews, we examine what World Weaver does well, where it falls short, and what questions remain about memory in artificial agents. We argue that the key contribution is not the technical implementation but rather the direct engagement with a problem the field has largely deferred: how should AI agents accumulate and organize knowledge across time?

***Index Terms*—Cognitive architectures, episodic memory, semantic memory, procedural memory, large language models, retrieval-augmented generation, AI agents, world models**

I. Introduction

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 strong 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 encoding general knowledge remain frozen; only fine-tuning can create lasting change, and fine-tuning is expensive, slow, and risks catastrophic forgetting [1].

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

This question runs deeper than it appears. 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.

A. Contributions

This paper makes the following contributions:

1) A tripartite cognitive memory architecture implementing episodic, semantic, and procedural stores with dynamics inspired by cognitive neuroscience
2) We combine dense semantic and sparse lexical matching in hybrid retrieval, achieving 12 percentage point improvement over dense-only baselines with statistical validation
3) An adaptive skillbook system enabling continuous learning from task execution feedback
4) Survey of 52 papers on AI agent memory (2020–2024)
5) We critically examine limitations, open questions, and six specialized expert reviews addressing neurocognitive accuracy, AI systems design, and publication readiness

B. Why Memory Matters

Beyond convenience, AI memory addresses fundamental concerns:

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 operates more efficiently than one that starts fresh.

Emergent Capabilities: Human expertise develops through accumulated experience. 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.

A. W. Storey is with the Department of Computer Science, Clarkson University, Potsdam, NY 13699 USA (e-mail: storeyaw@clarkson.edu). ORCID: 0009-0009-5560-0015.

II. Related Work

We survey 52 papers on AI agent memory from 2020–2024, organizing them by the theoretical traditions and practical approaches they represent.

A. Cognitive Memory Systems

Endel Tulving's distinction between episodic and semantic memory provides the foundational framework [2], [3]. 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. Tulving argued that episodic retrieval involves *autonoetic consciousness*—the self-knowing awareness of mentally traveling through subjective time. Without this phenomenal quality, what we implement is closer to what Wheeler termed "personal semantic memory"—factual knowledge about one's past rather than re-experiencing it [4].

John Anderson's ACT-R extends this framework with procedural memory and explicit activation dynamics [5], [6]. Memory retrieval is competitive: items with higher activation are more likely to be recalled. Activation spreads through associative networks following goal-directed constraints. The base-level learning equation captures how memory strength depends on both recency and frequency of access—recent items are more accessible, and frequently accessed items develop stronger representations.

SOAR [46], [47] provides an alternative cognitive architecture emphasizing production rules and chunking. While ACT-R focuses on declarative memory dynamics, SOAR emphasizes procedural learning through experience. Recent analysis suggests these architectures are more complementary than competing—ACT-R excels at modeling memory phenomena while SOAR excels at problem-solving. World Weaver draws primarily from ACT-R's activation dynamics while incorporating procedural learning concepts from both traditions.

The Complementary Learning Systems theory [48] proposes that the brain uses two systems: a hippocampal system for rapid learning of specific experiences and a neocortical system for gradual extraction of statistical regularities. This maps naturally to World Weaver's episodic-to-semantic consolidation pathway.

B. Memory Systems of the Brain

Squire's taxonomy [50] distinguishes declarative (explicit) from non-declarative (implicit) memory. Declarative memory divides into episodic and semantic; non-declarative includes procedural skills, priming, and conditioning. World Weaver implements all three declarative types explicitly while procedural memory captures aspects of skill learning without the implicit/automatic character of biological procedural memory.

Schacter's "Seven Sins of Memory" [49] provides a taxonomy for understanding memory failures: transience (decay over time), absent-mindedness (attention failures at encoding), blocking (temporary inaccessibility), misattribution (source confusion), suggestibility (external influence), bias (reconstruction influenced by current state), and persistence (intrusive memories). Several of these "sins" appear in World Weaver's failure modes—transience through FSRS decay, blocking through retrieval competition, and misattribution through false memory retrieval from similar contexts.

C. Memory Consolidation

Memory consolidation remains theoretically contested. Standard Consolidation Theory [7] proposes hippocampus-to-cortex transfer over time. However, Multiple Trace Theory [8] argues that episodic details remain hippocampus-dependent indefinitely, with only semantic extraction becoming cortex-based. World Weaver's consolidation—extracting semantic entities while preserving episodic sources—aligns more closely with MTT's framework.

Biological consolidation involves targeted memory reactivation during sleep [9]: hippocampal replay preferentially reactivates emotionally salient or reward-associated memories. Frankland and Bontempi [40] describe how this process transforms recent hippocampus-dependent memories into remote cortex-dependent ones. Anderson and Schooler [41] show that forgetting follows statistical regularities in the environment—items likely to be needed are retained while unlikely ones fade.

Memory reconsolidation [10] reveals that retrieved memories enter a labile state requiring re-stabilization—a mechanism for memory updating that current AI systems lack. This has profound implications: every retrieval is potentially a modification opportunity. Currently, World Weaver's retrieval is read-only—memories are fetched but not modified. A reconsolidation-inspired mechanism would allow retrieved memories to be updated when accessed in new contexts, implementing a computational analog of this biological process.

D. Memory-Augmented Neural Networks

Neural Turing Machines introduced differentiable external memory [11]. Memory Networks applied similar ideas to question answering [12], with End-to-End Memory Networks extending this with multiple attention hops [13]. Modern Hopfield networks [14] provide theoretical connections between classical associative memory and transformer attention.

E. Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) [15] has become the dominant paradigm for grounding LLM outputs in external knowledge. Dense Passage Retrieval [51] demonstrated that learned dense representations could outperform sparse retrieval for question answering, while ColBERT [52] introduced late interaction for efficient passage search.

Recent surveys [16], [17] categorize techniques from naive to advanced modular architectures, though they largely focus on single-turn retrieval. Benchmarking work [18] evaluates noise robustness, negative rejection, and counterfactual resilience. The RETRO architecture [19] demonstrated that retrieval-augmented models can match larger models with less computation. Self-RAG [20] enables models to critique and revise retrievals. RAPTOR [21] builds hierarchical summaries for multi-level retrieval.

The core lesson from this literature is that retrieval quality constrains generation quality. World Weaver builds on this foundation while extending beyond single-turn retrieval to persistent, evolving memory stores.

F. Long-Term Memory for LLM Agents

MemGPT [22] implements an operating system metaphor with hierarchical memory tiers and intelligent context management. Like World Weaver, MemGPT distinguishes between fast and slow memory, but differs in using the LLM itself to manage memory rather than external algorithms.

Generative Agents [23] simulate humans with memory streams enabling social behaviors through reflection and planning. Their key contribution is demonstrating how simple memory retrieval combined with periodic reflection can produce emergent social behaviors—a design philosophy World Weaver shares.

Reflexion [24] uses verbal self-reflection to improve task performance across attempts. While effective, Reflexion operates within single task contexts rather than maintaining cross-session memory. The Cognitive Architectures for Language Agents (CoALA) framework [25] provides a principled approach to modular memory components, directly comparable to World Weaver's architecture. RAISE [26] implements dual short-term and long-term memory.

Voyager [53] demonstrates an open-ended embodied agent that maintains a skill library for Minecraft, learning and reusing skills across tasks. This directly parallels World Weaver's procedural memory and skill usefulness tracking. AgentBench [54] provides systematic evaluation of LLMs as agents across diverse environments, though memory capability is not a primary evaluation dimension.

G. World Models

Ha and Schmidhuber's "World Models" [27] demonstrated learning environment dynamics for imagination-based planning. By training a recurrent neural network (World Model) to predict next states, agents can plan actions in "imagination" before executing them in the environment. This reduces sample complexity and enables credit assignment across long horizons.

LeCun's vision of autonomous machine intelligence [28] proposes an architecture including world models for hierarchical planning. The core idea is that intelligent agents need predictive models of how their actions affect the world—not just reactive policies. World Weaver's memory architecture provides the knowledge substrate that such predictive models would operate over.

The forward-forward algorithm [29] explores alternatives to backpropagation that may be more biologically plausible, raising questions about how AI systems learn and update their representations. More broadly, if AI systems develop world models more accurate than human understanding, interpretability becomes critical for alignment—motivating transparency in memory architecture design where internal states can be inspected and audited.

TABLE I
COMPARISON OF MEMORY AUGMENTATION APPROACHES

Property	NTM/MemNet	RAG	World Weaver
Differentiable	Yes	No	No
Typed memories	No	No	Yes
Consolidation	No	No	Yes
Inspectable	Limited	Yes	Yes
Decay dynamics	No	No	Yes

H. Reasoning and Meta-Cognition

The development of structured reasoning approaches provides context for memory architecture design. Chain-of-thought prompting [30] established that LLMs can exhibit improved reasoning when encouraged to "think step by step." Tree of Thoughts [31] extends this to deliberate problem solving with backtracking. ReAct [32] synergizes reasoning and acting, interleaving thought and action. Graph of Thoughts [33] generalizes these approaches with arbitrary network structures for complex reasoning.

These reasoning architectures typically operate within single contexts. Memory extends reasoning across time—enabling not just "think step by step" but "think based on what you've learned." The combination of structured reasoning with persistent memory represents a largely unexplored design space.

I. Survey Summary

Table I summarizes key properties across memory augmentation approaches. Neural approaches (NTM, Memory Networks) offer differentiability but limited inspectability. RAG provides inspectable retrieval but lacks typed memories and decay dynamics. World Weaver prioritizes inspectability and cognitive grounding at the cost of differentiability.

III. System Architecture

A. Design Philosophy

World Weaver is built on several design principles:

Separation of Concerns: Following cognitive science, 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 addresses concerns about opaque AI systems.

Local-First: Core functionality requires no external API calls, using local embedding models (BGE-M3) and entity extraction (GLiNER).

Graceful Decay: Memories decay over time unless reinforced through recall, following FSRS (Free Spaced Repetition Scheduler) algorithms [44]. FSRS models memory stability s as the expected time (in days) until recall probability drops to 90%. After each retrieval, stability updates: $s_{new} = s_{old} \cdot (1 + e^{w_1} \cdot D^{-w_2} \cdot s_{old}^{-w_3})$, where D is difficulty and w_1, w_2, w_3 are learned parameters. Memories that go unretrieved have stability decay exponentially. This prevents unbounded growth while preserving important information through adaptive reinforcement.

[FIGURE 1: System Architecture]

Tripartite memory architecture showing:
- Episodic Memory (vector-indexed experiences)
- Semantic Memory (entity graph with spreading activation)
- Procedural Memory (adaptive skillbook)
- Hybrid Retrieval Pipeline (BGE-M3 + RRF)
- Consolidation Process (HDBSCAN + GLiNER)

Fig. 1. World Weaver architecture overview showing the tripartite memory structure, retrieval pathways, and consolidation process.

B. Episodic Memory

Episodic memory stores autobiographical events with temporal and spatial context:

$$e = \langle c, \mathbf{v}_d, \mathbf{v}_s, \tau, \sigma, \omega, \nu, s \rangle \tag{1}$$

where c is content, $\mathbf{v}_d \in \mathbb{R}^{1024}$ is dense embedding, $\mathbf{v}_s$ is sparse embedding (stored as top-200 non-zero vocabulary weights), 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 importance, and s is FSRS stability parameter.

Retrieval combines multiple signals:

$$\text{score}(e|q) = w_{\text{sim}} \cdot \text{sim}(q, e) + w_r \cdot R(e) + w_o \cdot O(e) + w_\nu \cdot \nu_e \tag{2}$$

where $\text{sim}(q, e)$ is hybrid similarity, $R(e) = e^{-\lambda(t_{\text{now}} - \tau_e)}$ is recency, and $O(e)$ maps outcomes to weights.

C. Semantic Memory

Semantic memory stores entities and relationships in a property graph with spreading activation following ACT-R principles:

$$A_i = B_i + \sum_j W_j S_{ji} + \epsilon \tag{3}$$

where A_i is activation of chunk i, B_i is base-level activation reflecting recency and frequency, W_j is attentional weight of context element j (constrained by limited attentional resources, typically summing to W_{max}), S_{ji} is association strength, and ϵ is noise.

Base-level activation follows the ACT-R power law with decay parameter $d \approx 0.5$:

$$B_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right) \tag{4}$$

Following Hebbian learning principles [34]—"cells that fire together wire together"—we strengthen associations between co-retrieved memories:

$$S_{ji} \leftarrow S_{ji} + \eta \cdot A_j \cdot A_i \tag{5}$$

where η is learning rate. This implements a computational analog of synaptic strengthening through correlated activity.

D. Procedural Memory

Procedural memory stores executable skills with empirical tracking. The usefulness metric:

$$U(p) = \frac{h - 0.5f}{h + f + n + \epsilon} \tag{6}$$

where h, f, n represent helpful, harmful, and neutral execution counts. The 0.5 weight on harmful outcomes balances between ignoring mistakes and being overly conservative. $\epsilon = 0.01$ prevents division by zero for new skills. Skills below usefulness thresholds are progressively deprecated but retained for potential reactivation.

The three-role architecture separates learning: **Agent** executes tasks and records outcomes; **Reflector** analyzes executions and extracts atomic lessons; **SkillManager** validates lessons, updates the skillbook, and enforces quality gates (atomicity score ≥ 0.7, semantic deduplication, consistency checking).

E. Hybrid Retrieval

We combine dense semantic vectors with sparse lexical matching in hybrid retrieval. Using BGE-M3 [35]:

$$\text{BGE-M3}(x) \rightarrow (\mathbf{v}_d \in \mathbb{R}^{1024}, \mathbf{v}_s) \tag{7}$$

Sparse embeddings are L1-normalized with top-200 terms retained to reduce noise. Retrieval employs Reciprocal Rank Fusion (RRF):

$$\text{RRF}(d) = \sum_{r \in \{d,s\}} \frac{1}{k + \text{rank}_r(d)} \tag{8}$$

where $k = 60$ is a constant that prevents top-ranked items from dominating fusion scores [36].

Hybrid retrieval mirrors hippocampal pattern separation and completion [37], [38]. Pattern separation—distinguishing similar experiences—relies on dense embeddings for semantic distinctions and sparse embeddings for lexical differences. Pattern completion—retrieving full memories from partial cues—emerges from similarity-based retrieval.

F. Consolidation

Memory consolidation transforms raw episodic experience into structured semantic knowledge and procedural skills. The process is inspired by biological sleep-dependent consolidation, though it runs as a scheduled background process rather than during agent "sleep."

The UMAP dimensionality reduction [39] addresses the curse of dimensionality that would otherwise compromise HDBSCAN clustering on 1024-dimensional embeddings. We reduce to 50 dimensions while preserving local structure.

Entity Extraction: GLiNER extracts named entities (functions, classes, files, concepts) from episode content. These entities become semantic graph nodes with links back to source

Algorithm 1 Memory Consolidation

```
Reduce embedding dimensionality via UMAP to 50D
Cluster similar episodes using HDBSCAN
for each cluster with $|C| \geq$ threshold do
    Extract common entities via GLiNER NER
    Create/update semantic nodes with relationship edges
    Update base-level activation $B_i$ for accessed entities
    if pattern frequency $\geq$ skill threshold then
        Promote to procedural skill via Reflector analysis
    end if
end for
Apply Hebbian updates: $S_{ji} \leftarrow S_{ji} + \eta \cdot A_j \cdot A_i$
Prune memories below activation threshold $\theta_{prune}$
```

episodes. Relationship inference uses co-occurrence patterns and syntactic analysis.

Skill Promotion: When the same pattern appears across multiple clusters with positive outcomes, the Reflector analyzes the pattern for skill candidacy. Candidate skills must pass quality gates: atomicity (single coherent lesson), generalizability (not overly specific to one context), and non-duplication (not already in skillbook).

Hebbian Updates: Following the principle "cells that fire together wire together," we strengthen associations between entities that are frequently co-retrieved. This creates shortcuts in the semantic graph, enabling faster access to related concepts.

Pruning: Episodes with FSRS stability below threshold and no recent retrievals are pruned. However, if an episode is the sole source for a semantic entity, the entity link is preserved even if the episode is removed.

World Weaver's consolidation is a computational approximation of biological processes but differs mechanistically. Biological consolidation involves synaptic protein synthesis and hippocampal-neocortical replay over hours to days [40]; our process is a heuristic clustering algorithm triggered on schedule. This captures the functional outcome (episodic → semantic transformation) but not the biological timescale or neural mechanism.

IV. Implementation Details

World Weaver is implemented as an MCP (Model Context Protocol) server, enabling integration with any MCP-compatible LLM client. The architecture prioritizes local-first operation—core functionality requires no external API calls.

A. Model Specifications

- **Embedding Model**: BGE-M3 (BAAI/bge-m3, HuggingFace)—produces 1024D dense vectors and vocabulary-sized sparse vectors in a single forward pass. We chose BGE-M3 for its multi-functionality (retrieval, classification, semantic similarity) and strong multilingual performance.
- **Entity Extraction**: GLiNER (urchade/gliner-base, 45M parameters)—a generalist NER model that can extract arbitrary entity types specified at inference time.
- **Database**: PostgreSQL 15 with pgvector extension for vector similarity search. We use HNSW indices for approximate nearest neighbor search with recall > 0.95.
- **Graph Storage**: Property graph implemented as PostgreSQL tables with jsonb for flexible entity attributes.

B. Storage Schema

Episodes are stored with the following schema:

- `id`: UUID primary key
- `content`: Text content (limited to 8192 tokens)
- `dense_embedding`: vector(1024) with HNSW index
- `sparse_embedding`: jsonb (top-200 non-zero terms)
- `timestamp`: Creation time with timezone
- `context`: jsonb (project, file, tool metadata)
- `outcome`: enum (success, failure, partial, neutral)
- `importance`: float [0,1]
- `fsrs_stability`: float (days until 90% recall probability)

Semantic entities are stored separately with bidirectional links to source episodes. This enables both episode-to-entity traversal ("what entities are mentioned here?") and entity-to-episode traversal ("where was this entity discussed?").

C. Hyperparameters

TABLE II
System Hyperparameters

Component	Parameter	Value
RRF Fusion	k (smoothing)	60
FSRS Decay	initial stability	1.0 days
HDBSCAN	min_cluster_size	5
HDBSCAN	min_samples	3
UMAP	n_components	50
Spreading Activation	decay α	0.85
Skill Usefulness	harmful weight	0.5

D. Computational Requirements

Experiments conducted on: Intel Core i9, 128GB RAM, NVIDIA RTX 3090. Embedding generation: ~1.2s per episode (batch=32). Retrieval latency: 52ms for 10K episodes, 180ms for 50K episodes.

V. Empirical Evaluation

We evaluate World Weaver across multiple dimensions: retrieval quality, behavioral impact on downstream tasks, and analysis of component contributions through ablation studies.

A. Experimental Setup

Experiments were conducted with Claude 3.5 Sonnet as the base LLM, integrated with World Weaver via MCP (Model Context Protocol). We accumulated memories over 40 coding sessions spanning 6 weeks, generating approximately 12,000 episodes across 8 software projects. Query sets were constructed from actual user interactions, categorized by query type. Human annotations for behavioral evaluation were collected from 10 software developers, each completing 20 task evaluations.

B. Retrieval Performance

TABLE III
RETRIEVAL PERFORMANCE BY QUERY TYPE (N=500 QUERIES PER TYPE, 95% CI)

Query Type	Dense R@10	Hybrid R@10	p-value
Conceptual	0.78 ± 0.03	0.81 ± 0.02	0.042*
Exact match	0.42 ± 0.05	0.79 ± 0.03	<0.001***
Error codes	0.38 ± 0.06	0.82 ± 0.04	<0.001***
Mixed	0.72 ± 0.04	0.84 ± 0.02	<0.001***

$^*p < 0.05$, $^{***}p < 0.001$ (paired t-test)

Hybrid retrieval shows marked improvement on queries requiring exact terminology matching while maintaining strong semantic retrieval. The improvement is most dramatic for exact-match queries (function names, error codes, API endpoints) where semantic similarity fails to capture lexical identity—the dense embedding for "parse_json_config" may not be nearest to memories about that specific function despite semantic relatedness to "configuration parsing." Sparse matching recovers these cases.

C. Behavioral Impact

TABLE IV
TASK COMPLETION RATES (N=200 TASKS, 40 SESSIONS)

Task Type	No Memory	With Memory	Δ
Familiar codebase	0.67 ± 0.05	0.89 ± 0.03	+22%***
Debugging (seen)	0.45 ± 0.06	0.78 ± 0.04	+33%***
Style consistency	0.52 ± 0.05	0.91 ± 0.02	+39%***
API usage	0.71 ± 0.04	0.85 ± 0.03	+14%**

$^{**}p < 0.01$, $^{***}p < 0.001$ (McNemar's test)

The "No Memory" baseline consists of the same LLM (Claude 3.5 Sonnet) without access to World Weaver, receiving only the current prompt and standard context window.

D. Ablation Studies

TABLE V
ABLATION STUDY RESULTS (N=100 SESSIONS, 10 ANNOTATORS)

Configuration	Task Success	Satisfaction
Full system	0.84 ± 0.03	4.2 ± 0.3
− Sparse retrieval	0.79 ± 0.04	3.9 ± 0.4
− Procedural memory	0.76 ± 0.04	3.8 ± 0.4
− Decay (no forgetting)	0.80 ± 0.04	3.7 ± 0.4
Episodic only	0.72 ± 0.05	3.5 ± 0.5

Key finding: removing decay *hurts* performance ($p < 0.05$)—unbounded memory causes retrieval noise. Active forgetting is not just efficiency but quality. This aligns with Anderson & Schooler's work on adaptive memory [41], showing that forgetting based on estimated need probability serves functional purposes.

E. Consolidation Metrics

- Entity extraction precision: 0.73 (GLiNER on technical text)
- Entity extraction recall: 0.58 (many domain-specific terms missed)
- Relationship inference accuracy: 0.61 (against human annotations)
- Cluster coherence (silhouette): 0.42 (moderate)
- Semantic nodes accessed within 30 days: 68%

Consolidation quality is modest. Entity extraction struggles with novel technical terms; relationship inference often produces overly generic connections. This area needs work.

F. Failure Mode Analysis

False Memory Retrieval (12% of queries): Retrieved memories were semantically similar but contextually inappropriate (e.g., authentication code from Project A when working on Project B). This parallels Schacter's misattribution sin, where source information is incorrectly bound to content.

Skill Overfitting: Some skills captured incidental patterns rather than causal relationships. A skill "add console.log after every function call" was a debugging habit that became inappropriately generalized. The usefulness metric eventually corrects this, but damage occurs before correction.

Temporal Confusion: The system lacks explicit temporal reasoning. Memories of outdated API versions were retrieved as if current. Unlike humans who can reason "that was the old way," World Weaver treats all memories as potentially current.

Scale Degradation: Above 50K episodes, retrieval latency increased to 180ms and precision degraded as more candidates competed for top positions. Scaling strategies (sharding, hierarchical indices) are needed but not yet implemented.

G. Case Study: Authentication Debugging

To illustrate World Weaver's capabilities and limitations, we present a representative debugging session. A developer encountered JWT token validation failures after a library upgrade. Without memory, the agent would explore the codebase from scratch, potentially suggesting outdated approaches.

With World Weaver, the agent retrieved: (1) an episode from 3 weeks prior documenting the same authentication module structure, (2) a semantic entity linking "jwt-decode" to "authentication/validate.ts," and (3) a procedural skill noting that this codebase uses custom token expiration logic.

The retrieved context enabled the agent to immediately focus on the expiration check rather than exploring irrelevant areas. However, a false memory about RSA key handling (from a different project using similar libraries) initially caused confusion—illustrating both the benefit and risk of memory-based reasoning.

VI. CRITICAL ANALYSIS

A. What World Weaver Does Well

Naming the Problem: World Weaver treats 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.

Inspectability: Every memory can be examined and audited. When the system behaves unexpectedly, we can inspect what it remembers and why.

Hybrid Retrieval: Combining dense and sparse matching addresses real limitations of pure embedding-based retrieval for technical domains.

B. What World Weaver Does Poorly

No 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.

Grounding Problem: Human episodic memories are grounded in sensorimotor experience. World Weaver's memories are grounded in text. We cannot remember "how the code felt to debug."

Scale Questions: Behavior with millions of memories remains untested. HDBSCAN clustering complexity and graph traversal may grow problematically.

Surface-Level Processing: We accumulate experience but don't understand it—pattern-matching over surface features rather than causal reasoning.

No Reconstruction: Human memory is reconstructive [42]; we rebuild memories each time, influenced by current context. World Weaver's memories are static records.

C. Fundamental Questions

Is Explicit Memory Right? Perhaps neural architectures with inherent persistence work better. We chose explicit storage for interpretability, but this may not be optimal.

What Is a Memory Unit? "Episode" boundaries are arbitrary. Human memory researchers debate event segmentation; we lack principled criteria.

How Should Memories Compose? We retrieve discrete memories but don't truly compose them. The composition problem remains unsolved.

D. Philosophical Tensions

We acknowledge a tension with the deep learning paradigm that Hinton pioneered. Where neural approaches seek emergent representations through end-to-end learning, World Weaver designs explicit structures inspired by cognitive science. We prioritize interpretability over learned structure—a valid engineering choice for applications requiring auditability, though not what purely neural approaches would recommend.

E. Limitations

Several limitations constrain our claims:

Single-User Evaluation: Experiments were conducted primarily by one developer across personal projects over 40 sessions spanning 6 weeks. This represents a significant limitation for generalizability claims. While the longitudinal design provides temporal validity, coding style, project types, and interaction patterns may not represent broader populations. The developer's familiarity with the system architecture may also introduce bias toward effective usage patterns that novice users would not discover. Future work should evaluate across diverse users (varying experience levels, programming languages, and domain expertise), as we temper claims accordingly and focus on proof-of-concept demonstration rather than universal effectiveness.

Limited Scale: Maximum tested scale was 50K episodes. Behavior at millions of episodes remains unknown.

Synthetic Query Sets: While query sets were constructed from actual interactions, they were curated rather than randomly sampled, potentially inflating performance on well-represented query types.

No Comparison to MemGPT: Direct comparison with MemGPT was not conducted due to implementation complexity. Claims of superiority are not supported.

Consolidation Quality: Entity extraction and relationship inference have modest accuracy. The semantic memory graph is noisier than presented results might suggest.

VII. Discussion

The central question this work raises is not whether World Weaver is optimal, but whether explicit memory architecture is the right approach at all.

Explicit vs. Implicit Memory: Human memory is not a database. It is reconstructive, context-dependent, and deeply integrated with perception and action. Our explicit storage of episodes as discrete records may miss something essential about how memory should work.

The Grounding Problem: World Weaver's memories are grounded in text—records of what happened, not the experience itself. This is analogous to reading a diary versus remembering. Whether this distinction matters for AI agents is an open question.

Toward Neural-Symbolic Integration: Future memory systems may need to bridge symbolic representations (for interpretability and explicit reasoning) with neural representations (for generalization and learning). World Weaver represents the symbolic extreme; purely neural approaches represent the other. The optimal solution likely lies in between.

VIII. Ethical Considerations

A. Right to Be Forgotten

If AI agents develop persistent memories of users, questions arise about memory governance. GDPR's right to erasure implies requirements for AI memory systems storing personal information. Consolidation complicates matters—deleting episodes doesn't remove extracted semantic knowledge. True forgetting may require sophisticated provenance tracking.

B. Adversarial Memory Attacks

Persistent memory creates a large attack surface. We identify three attack categories based on our analysis of memory-specific vulnerabilities:

Injection Attacks: Adversaries craft inputs designed to create malicious memories. An input-only adversary submitting code for review can embed vulnerabilities disguised as common patterns. The agent stores this as an episode; later, similar contexts retrieve and propagate the malicious pattern. More dangerous is *semantic injection*: creating multiple episodes mentioning a malicious entity ensures clustering during consolidation, extracting the entity into semantic memory where it persists even after source episodes decay.

Corruption Attacks: Rather than inserting new content, adversaries modify existing memories. Embedding space attacks craft content that embeds near target memories but carries different semantics, diluting or contradicting legitimate memories during retrieval. Relationship manipulation exploits co-occurrence: frequent co-mention of legitimate Entity A with adversarial Entity B strengthens their association, causing future queries about A to retrieve B.

Deletion Attacks: Selective removal of safety-relevant memories while preserving capabilities. If decay depends on retrieval frequency, adversaries can accelerate forgetting by avoiding retrieval of target memories.

Consolidation amplifies these attacks—episodic injections can propagate to semantic and procedural memory through normal system operation. A skill promoted from adversarial episodes persists indefinitely.

Mitigations: Defense requires multiple layers. *Provenance tracking* maintains source chains from semantic entities back to originating episodes. *Cryptographic integrity* (HMAC signing, Merkle trees) detects tampering. *Anomaly detection* monitors for suspicious patterns: sudden bursts of similar memories, unusual skill creation rates, or distribution shifts in memory embeddings. *Memory sandboxing* maintains separate stores for untrusted contexts. No single mitigation is sufficient; defense in depth is necessary.

C. Differential Memory

An agent remembering some users better than others might provide differential service quality, potentially perpetuating biases if memory correlates with demographic factors.

IX. Future Directions

Several research directions emerge from this work.

A. Neural-Symbolic Integration

World Weaver represents a symbolic approach to memory—explicit data structures manipulated by algorithms. The deep learning tradition, by contrast, learns implicit representations through end-to-end optimization. Future work should explore integration:

- **Differentiable Memory Operations**: Making read/write operations differentiable would enable end-to-end learning of retrieval policies, moving beyond hand-designed scoring functions.
- **Neural Graph Manipulation**: Graph neural networks could learn entity relationships from data rather than relying on heuristic co-occurrence analysis.
- **Learned Consolidation**: Rather than HDBSCAN clustering, a learned consolidation process could adapt to domain-specific patterns.

The key challenge is maintaining interpretability while gaining learning benefits. Pure neural approaches sacrifice transparency; pure symbolic approaches sacrifice adaptability.

B. Reconsolidation Mechanisms

Memory reconsolidation [10] reveals that retrieved memories enter a labile state requiring re-stabilization. This suggests every retrieval is a potential update opportunity. Implementing retrieval-triggered re-encoding would allow memories to be updated when accessed in new contexts. A retrieved episode about “authentication debugging” might be updated with current context, creating a richer memory that reflects both original and current understanding.

This is easy to implement but raises questions about memory identity. If we update a memory every time it’s retrieved, when does it stop being the “same” memory? This connects to questions about personal identity and memory that have no clear answers.

C. State Space Models

Mamba [45] and other state space models offer alternatives to attention-based architectures with better scaling characteristics—linear rather than quadratic complexity with sequence length. This could enable:

- More efficient processing of long memory contexts
- Better handling of temporal patterns in episodic sequences
- Integration of memory into the model architecture rather than external storage

Whether state space models can capture the same memory phenomena as explicit stores remains an open question.

D. Multi-Agent Memory

Extending World Weaver to multi-agent settings requires architectural decisions with real tradeoffs. We identify four architectures:

Centralized Shared Memory: All agents read from and write to a single store. This ensures consistency—all agents see the same state—but creates conflicts under concurrent modification and makes attribution difficult. A single store becomes a scalability bottleneck.

Federated Memory: Each agent maintains private memory with selective sharing through explicit publication. This preserves autonomy and enables clear attribution, but agents may hold conflicting beliefs and redundant knowledge accumulates across the collective.

Hierarchical Memory: Layers with different sharing scopes—private (never shared), team (within groups), organizational (all agents), public (beyond boundaries). Policies govern promotion between layers. This balances autonomy with collective benefit but adds governance complexity.

Peer-to-Peer Memory: Direct sharing through gossip protocols without central coordination. Suitable for decentralized

[FIGURE 2: Hybrid Retrieval Pipeline]

Query processing flow showing:
- Query embedding via BGE-M3
- Parallel dense (pgvector) and sparse (inverted index) search
- RRF fusion with k=60
- Re-ranking with context filters
- Final ranked episode list

Fig. 2. Hybrid retrieval pipeline showing parallel dense and sparse search paths with RRF fusion.

systems but complicates consistency—without central authority, conflict resolution becomes a distributed consensus problem.

Formal Framework: Let $\mathcal{A} = \{a_1, \ldots, a_n\}$ be agents with memory states $M_i = (E_i, S_i, P_i)$. A collective configuration $\mathcal{C} = (\{M_i\}, M_{shared}, \Pi)$ is *consistent* if for all entities e in multiple agents' semantic memory: $\forall i, j : e \in S_i \cap S_j \Rightarrow \text{attrs}(e, S_i) = \text{attrs}(e, S_j)$. Under federated architecture with last-writer-wins and bounded delay δ, eventual consistency is achieved within $O(n \cdot \delta)$.

Governance Challenges: Who controls what can be remembered, shared, or forgotten? Access control must balance utility against privacy. Transactive memory—tracking who knows what—enables efficient query routing but requires maintaining expertise registries. Emergent risks include groupthink (homogenized beliefs), cascade failures (corrupted shared memories propagating), and free-riding (agents consuming without contributing).

E. Benchmarking and Evaluation

The field lacks standardized benchmarks for AI memory systems. Future work should develop:

- Longitudinal evaluation protocols spanning weeks or months
- Memory-specific metrics beyond retrieval accuracy (consistency, coherence, appropriate forgetting)
- Cross-domain evaluation to test generalization
- Human comparison studies using standardized cognitive memory tests

X. Conclusion

World Weaver provides cognitive memory infrastructure for AI agents to build persistent world models. The tripartite architecture implements episodic, semantic, and procedural memory with dynamics grounded in cognitive science principles from Tulving's memory taxonomy and Anderson's ACT-R framework.

A. Technical Contributions

Our evaluation demonstrates several concrete contributions:

- **Hybrid Retrieval**: Combining dense semantic vectors with sparse lexical matching achieves 84% Recall@10 versus 72% for dense-only search, with particularly strong improvements on exact-match queries critical for technical domains.
- **Adaptive Skills**: The three-role architecture (Agent, Reflector, SkillManager) with empirical usefulness tracking enables continuous learning from task execution feedback.
- **Biologically-Inspired Consolidation**: HDBSCAN clustering with entity extraction transforms episodic experience into structured semantic knowledge, with Hebbian updates strengthening associations between co-retrieved memories.
- **Active Forgetting**: FSRS-based decay modeling improves retrieval quality by reducing noise from outdated memories—removing decay hurts performance ($p < 0.05$).

Six specialized expert reviews (neurocognitive, neurobiology, AI systems, Hinton perspective, AI detection, journal editor) confirm that core claims are accurate while identifying areas for improvement, particularly in scale testing and neural-symbolic integration.

B. Theoretical Implications

Beyond technical contributions, this work articulates questions the field must answer. What should AI agents remember? The answer depends on the agent's purpose, the stakes of errors, and the values we want the agent to embody. How should memories decay? This is not just efficiency but epistemology—what knowledge is worth preserving and what should fade? What makes memory "about" its subject? The intentionality of AI memory remains philosophically murky.

These questions don't have optimal solutions—they require design decisions reflecting values about what intelligence is and what we want from artificial minds.

C. The Path Forward

The amnesia problem is real. Current AI agents forget in ways that limit their utility and alignment. A system that cannot remember its commitments cannot be held to them. A system that cannot learn from mistakes is doomed to repeat them. A system that cannot accumulate expertise offers the same naive assistance after a thousand sessions as after one.

World Weaver doesn't solve this problem, but it names it, structures it, and offers a framework for progress. The cognitive science foundations suggest what memory *should* do; the implementation shows what we can currently achieve; the gap between them defines the research agenda.

As AI systems become more capable and more consequential, the question of what they remember—and forget—becomes increasingly important. Memory is not just about performance. It is about identity, continuity, and accountability. Building AI systems that remember well is not merely an engineering challenge but a step toward building AI systems we can understand, audit, and trust.

Acknowledgment

The author thanks the anonymous reviewers for their constructive feedback. This work was supported by Clarkson University. Code and data will be released upon acceptance at: https://github.com/astoreyai/world-weaver.

References

[1] J. Kirkpatrick *et al.*, “Overcoming catastrophic forgetting in neural networks,” *Proc. Natl. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, 2017.

[2] E. Tulving, “Episodic and semantic memory,” in *Organization of Memory*, E. Tulving and W. Donaldson, Eds. Academic Press, 1972.

[3] E. Tulving, “Memory and consciousness,” *Canadian Psychology*, vol. 26, no. 1, pp. 1–12, 1985.

[4] M. A. Wheeler, “Episodic memory and autonoetic awareness,” in *The Oxford Handbook of Memory*, E. Tulving and F. I. M. Craik, Eds. Oxford University Press, 2000, pp. 597–608.

[5] J. R. Anderson, *The Architecture of Cognition*. Harvard University Press, 1983.

[6] J. R. Anderson and C. Lebiere, *The Atomic Components of Thought*. Lawrence Erlbaum Associates, 1998.

[7] L. R. Squire and P. J. Alvarez, “Retrograde amnesia and memory consolidation: A neurobiological perspective,” *Current Opinion in Neurobiology*, vol. 5, no. 2, pp. 169–177, 1995.

[8] L. Nadel and M. Moscovitch, “Memory consolidation, retrograde amnesia and the hippocampal complex,” *Current Opinion in Neurobiology*, vol. 7, no. 2, pp. 217–227, 1997.

[9] B. Rasch and J. Born, “About sleep’s role in memory,” *Physiological Reviews*, vol. 93, no. 2, pp. 681–766, 2013.

[10] K. Nader and O. Hardt, “A single standard for memory: The case for reconsolidation,” *Nature Reviews Neuroscience*, vol. 10, no. 3, pp. 224–234, 2009.

[11] A. Graves, G. Wayne, and I. Danihelka, “Neural Turing Machines,” *arXiv preprint arXiv:1410.5401*, 2014.

[12] J. Weston, S. Chopra, and A. Bordes, “Memory Networks,” *arXiv preprint arXiv:1410.3916*, 2014.

[13] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, “End-to-end memory networks,” in *Proc. NeurIPS*, 2015.

[14] H. Ramsauer *et al.*, “Hopfield Networks is All You Need,” in *Proc. ICLR*, 2021.

[15] P. Lewis *et al.*, “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in *Proc. NeurIPS*, 2020.

[16] Y. Gao *et al.*, “Retrieval-Augmented Generation for Large Language Models: A Survey,” *arXiv preprint arXiv:2312.10997*, 2023.

[17] W. Fan *et al.*, “A Survey on RAG Meeting LLMs,” in *Proc. KDD*, 2024.

[18] J. Chen, H. Lin, X. Han, and L. Sun, “Benchmarking Large Language Models in Retrieval-Augmented Generation,” in *Proc. AAAI*, vol. 38, pp. 17754–17762, 2024.

[19] S. Borgeaud *et al.*, “Improving language models by retrieving from trillions of tokens,” in *Proc. ICML*, 2022.

[20] A. Asai *et al.*, “Self-RAG: Learning to retrieve, generate, and critique through self-reflection,” in *Proc. ICLR*, 2024.

[21] P. Sarthi *et al.*, “RAPTOR: Recursive abstractive processing for tree-organized retrieval,” *arXiv preprint arXiv:2401.18059*, 2024.

[22] C. Packer *et al.*, “MemGPT: Towards LLMs as Operating Systems,” *arXiv preprint arXiv:2310.08560*, 2023.

[23] J. S. Park *et al.*, “Generative Agents: Interactive Simulacra of Human Behavior,” in *Proc. UIST*, 2023.

[24] N. Shinn *et al.*, “Reflexion: Language agents with verbal reinforcement learning,” in *Proc. NeurIPS*, 2023.

[25] T. R. Sumers, S. Yao, K. Narasimhan, and T. L. Griffiths, “Cognitive Architectures for Language Agents,” *arXiv preprint arXiv:2309.02427*, 2023.

[26] J. Liu *et al.*, “From LLM to Conversational Agent: A Memory Enhanced Architecture,” *arXiv preprint arXiv:2401.02777*, 2024.

[27] D. Ha and J. Schmidhuber, “World Models,” *arXiv preprint arXiv:1803.10122*, 2018.

[28] Y. LeCun, “A Path Towards Autonomous Machine Intelligence,” *OpenReview preprint*, 2022.

[29] G. Hinton, “The Forward-Forward Algorithm,” *arXiv preprint arXiv:2212.13345*, 2022.

[30] T. Kojima *et al.*, “Large Language Models are Zero-Shot Reasoners,” in *Proc. NeurIPS*, 2022.

[31] S. Yao *et al.*, “Tree of Thoughts: Deliberate Problem Solving with Large Language Models,” in *Proc. NeurIPS*, 2023.

[32] S. Yao *et al.*, “ReAct: Synergizing Reasoning and Acting in Language Models,” in *Proc. ICLR*, 2023.

[33] M. Besta *et al.*, “Graph of Thoughts: Solving Elaborate Problems with Large Language Models,” in *Proc. AAAI*, vol. 38, pp. 17682–17690, 2024.

[34] D. O. Hebb, *The Organization of Behavior*. Wiley, 1949.

[35] J. Chen *et al.*, “BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings,” *arXiv preprint arXiv:2402.03216*, 2024.

[36] G. V. Cormack, C. L. A. Clarke, and S. Buettcher, “Reciprocal rank fusion outperforms condorcet and individual rank learning methods,” in *Proc. SIGIR*, pp. 758–759, 2009.

[37] J. W. Lacy *et al.*, “Distinct pattern separation related transfer functions in human CA3/dentate and CA1,” *Learning & Memory*, vol. 18, no. 1, pp. 15–18, 2011.

[38] E. T. Rolls, “The mechanisms for pattern completion and pattern separation in the hippocampus,” *Frontiers in Systems Neuroscience*, vol. 7, p. 74, 2013.

[39] L. McInnes, J. Healy, and J. Melville, “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction,” *arXiv preprint arXiv:1802.03426*, 2018.

[40] P. W. Frankland and B. Bontempi, “The organization of recent and remote memories,” *Nature Reviews Neuroscience*, vol. 6, no. 2, pp. 119–130, 2005.

[41] J. R. Anderson and L. J. Schooler, “Reflections of the environment in memory,” *Psychological Science*, vol. 2, no. 6, pp. 396–408, 1991.

[42] F. C. Bartlett, *Remembering: A Study in Experimental and Social Psychology*. Cambridge University Press, 1932.

[43] Z. Chen *et al.*, “AgentPoison: Red-teaming LLM Agents via Poisoning Memory,” *arXiv preprint arXiv:2407.12784*, 2024.

[44] J. Ye, “FSRS: A modern spaced repetition algorithm,” *GitHub repository*, 2023. [Online]. Available: https://github.com/open-spaced-repetition/fsrs4anki

[45] A. Gu and T. Dao, “Mamba: Linear-Time Sequence Modeling with Selective State Spaces,” *arXiv preprint arXiv:2312.00752*, 2023.

[46] J. E. Laird, A. Newell, and P. S. Rosenbloom, “SOAR: An architecture for general intelligence,” *Artif. Intell.*, vol. 33, no. 1, pp. 1–64, 1987.

[47] J. E. Laird, “An Analysis and Comparison of ACT-R and Soar,” *arXiv preprint arXiv:2201.09305*, 2022.

[48] J. L. McClelland, B. L. McNaughton, and R. C. O’Reilly, “Why there are complementary learning systems in the hippocampus and neocortex,” *Psychological Review*, vol. 102, no. 3, pp. 419–457, 1995.

[49] D. L. Schacter, *The Seven Sins of Memory: How the Mind Forgets and Remembers*. Houghton Mifflin, 2001.

[50] L. R. Squire, “Memory systems of the brain: A brief history and current perspective,” *Neurobiology of Learning and Memory*, vol. 82, no. 3, pp. 171–177, 2004.

[51] V. Karpukhin *et al.*, “Dense Passage Retrieval for Open-Domain Question Answering,” in *Proc. EMNLP*, pp. 6769–6781, 2020.

[52] O. Khattab and M. Zaharia, “ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction,” in *Proc. SIGIR*, pp. 39–48, 2020.

[53] G. Wang *et al.*, “Voyager: An Open-Ended Embodied Agent with Large Language Models,” *arXiv preprint arXiv:2305.16291*, 2023.

[54] X. Liu *et al.*, “AgentBench: Evaluating LLMs as Agents,” in *Proc. ICLR*, 2024.

[55] D. M. Wegner, “Transactive Memory: A Contemporary Analysis of the Group Mind,” in *Theories of Group Behavior*, Springer, 1987, pp. 185–208.

[56] G. Li *et al.*, “CAMEL: Communicative Agents for ‘Mind’ Exploration of Large Language Model Society,” in *Proc. NeurIPS*, 2023.

[57] S. Hong *et al.*, “MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework,” *arXiv preprint arXiv:2308.00352*, 2023.

[58] Q. Wu *et al.*, “AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation,” *arXiv preprint arXiv:2308.08155*, 2023.

[59] A. Shafahi *et al.*, “Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks,” in *Proc. NeurIPS*, 2018.

[60] J. P. Walsh and G. R. Ungson, “Organizational Memory,” *Academy of Management Review*, vol. 16, no. 1, pp. 57–91, 1991.