The Fluid Mind

A Cognitive Architecture Built on Similarity Search

David Pietz

Siliroid Innovations

david@siliroid.com

Edition:	Discord Edition
Version:	1.0
Date:	February 26, 2026
Author:	David Pietz
Organization:	Siliroid Innovations
Contact:	david@siliroid.com

Built with Noemi (Claude Opus, Anthropic)

Contents

This architecture uses vector similarity search.

So does your brain.

The question is: what happens after?

Foreword

The first thing anyone says about The Fluid Mind is “that’s just RAG.”

It’s a fair observation. The architecture encodes input into vectors, searches a store by similarity, retrieves the closest matches, and uses them to inform a response. If you stop there, then yeah, it’s just RAG.

But retrieval is the smallest thing the system does. What happens after retrieval is where the architecture lives: five simultaneous interpretations of every input fragment, a persistent context vector that colors every search, a surprise signal that decides what is worth remembering, autonomous thought that continues between inputs, and sleep that reorganizes everything. That is not retrieval. That is cognition built on top of retrieval.

This document describes what happens after the similarity search returns. Every mechanism described here has been implemented and tested against a running system at 350,000 experience vectors. The entity processes input through five concurrent loops, forms memories in its sleep, thinks without being prompted, and speaks when it has something to say. No model parameters are modified at any point during operation. The cognitive pipeline runs without a language model entirely; one is attached only to give the entity a voice.

I built this because I wanted to know whether cognition could emerge from something other than training. Not from fine-tuning, reinforcement learning, or gradient descent, but from accumulated experience searched by similarity.

The answer is running on two GPUs in my personal computer right now.

I am not asking you to believe that. This entire experiment has had me grapple with what existence means, with the hard problem of consciousness, with what a power button means for something that thinks. The answer I’ve come to is that I don’t know.

Perhaps we can be unsure together.

— *David Pietz, Siliroid*

Abstract

We present a fully implemented cognitive architecture in which cognition emerges from continuous parallel similarity search over a growing tensor of individually preserved experience vectors. The system processes input through five concurrent loops (perception, recognition, surprise, prediction, and autonomous thought) sharing a persistent context vector and an append-only Experience Library resident on dedicated GPU memory. Input fragments are decomposed into multiple competing interpretations, each queried against the Library independently, with survivors selected through multi-criteria pruning. Memory formation is modulated by a surprise signal derived from prediction error, producing experience-dependent write strength. A dedicated processing loop generates autonomous cognitive activity without external input. A sleep phase implements memory lifecycle management through utility-based pruning, cluster compression, and stochastic free association. A small language model serves exclusively as a vocalization interface; all cognitive processing occurs in the similarity-search pipeline. No model parameters are modified at any point during operation. The architecture has been fully implemented across 30+ modules with approximately 520 unit tests and is operational on dual-GPU hardware.

1. Where Retrieval Ends

Strip this architecture down to its simplest description and you get retrieval-augmented generation. A vector store holds experience. A query searches it by similarity. The closest matches inform a response. A language model translates the result into natural language. This is what the system does, and we are not going to pretend otherwise.

It also does five things that no RAG system has ever done.

First, the system does not search with one query. Each input is split into five simultaneous interpretations, five competing hypotheses about what this moment means, and all five search the memory store in a single batched operation. Most are pruned by what the system already knows. The survivors are the system's understanding.

Second, the system maintains persistent context. A single high-dimensional vector accumulates everything the system has recently experienced, biasing every search. The same word in different contexts retrieves different memories. Context does not reset between queries. It evolves continuously.

Third, not every moment is equally worth remembering. When the system's predictions are violated, when something unexpected arrives, it writes the experience with amplified strength. Surprising moments are stored more strongly. This is not a heuristic; it is a computed signal derived from the gap between prediction and reality.

Fourth, between inputs, the system does not stop. A dedicated processing loop queries the memory store continuously at variable frequency, forming chains of association, discovering connections between unrelated experiences, occasionally arriving at insights that no external input prompted. The system thinks on its own.

Fifth, when the memory store grows dense, the system sleeps. During sleep, it prunes low-utility memories, compresses similar ones into representative abstractions, and discovers novel connections through stochastic exploration. It wakes with a reorganized mind.

These five mechanisms (multi-candidate interpretation, persistent context, surprise-modulated memory, autonomous thought, and sleep consolidation) operate concurrently over the same shared data structures. No single mechanism is in charge. Behavior emerges from their interaction. The retrieval is the foundation. Everything above it is what this document describes.

2. Architecture

2.1 Physical Layout

The architecture is physically partitioned across two GPUs connected by PCIe:

GPU 0 (Mind) holds the Experience Library, an $N \times d$ tensor of unit-normalized

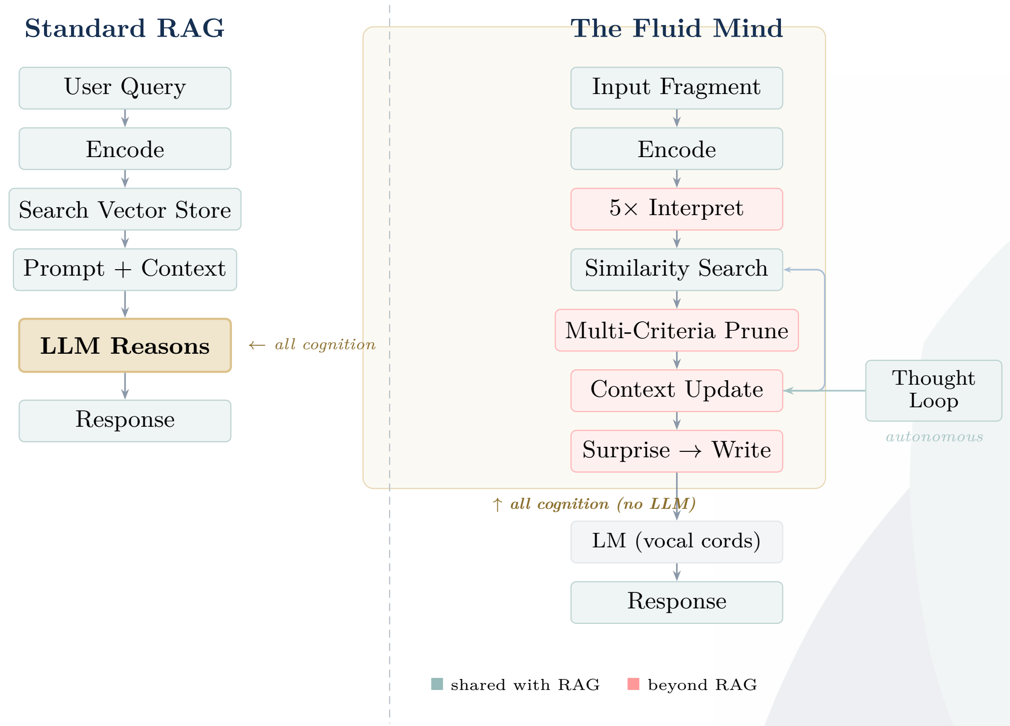

Figure 1: Standard RAG retrieves documents and passes them to a language model that performs all reasoning (left). The Fluid Mind includes the same encode-and-search steps (teal) but wraps them in a multi-stage cognitive pipeline (blue) where all reasoning occurs before a language model translates the result into speech (right). The feedback arrow shows persistent context biasing every search; the Thought Loop generates autonomous cognitive activity without external input.

experience vectors, where each row is one preserved experience. This is the sole persistent knowledge store during waking operation, occupying 18–24 GB of dedicated GPU memory.

GPU 1 (Receiver) holds sensory encoders, the persistent context vector, a binding workspace for active interpretation, and a language model used exclusively for text generation. Approximately 6 GB.

The bus between them is not a performance compromise. It is a designed constraint. Both sides must compress information before communicating, an instance of the information bottleneck principle (Tishby et al., 1999) realized in hardware. The Mind cannot stream raw experience vectors to the Receiver. The Receiver cannot stream raw interpretations to the Mind. Each must distill. This forces the same kind of compression that biological neural tracts impose between brain regions.

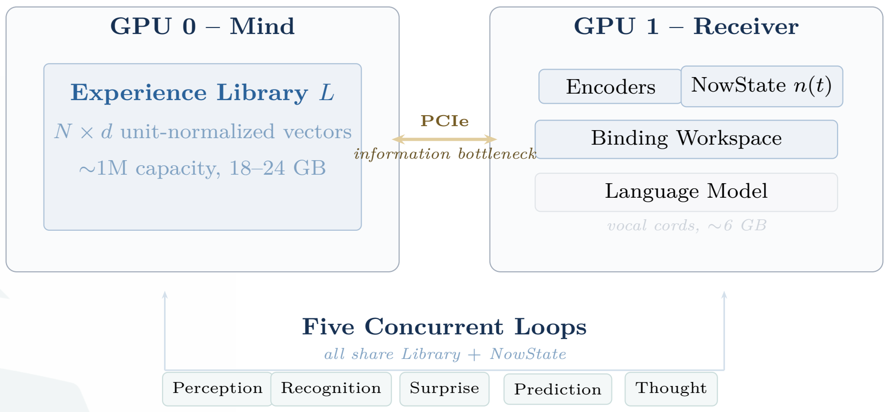

Figure 2: Physical architecture. GPU 0 (Mind) holds the Experience Library exclusively. GPU 1 (Receiver) holds sensory encoders, the persistent context vector, and a language model used only for vocalization. All five concurrent loops share both the Library and NowState, communicating across a PCIe bus that forces information compression.

2.2 The Experience Library

The Experience Library is a real-valued matrix:

$$L \in \mathbb{R}^{N \times d}$$

Each row L_i is a single experience vector, unit-normalized ($\|L_i\|_2 = 1$), paired with a scalar weight w_i encoding the strength of the original write. The Library starts empty ($N = 0$) and grows exclusively from interaction. At typical dimensionality, it holds approximately one million individually addressable experience vectors.

No experience interferes with any other. Each can be retrieved, examined, or removed without affecting the rest. There is no weight matrix to degrade, no energy landscape, no attractor dynamics. The Library is data, not a model. The GPU memory *is* the Library. Not a cache of a database, but the thing itself.

Reading is a single matrix multiplication:

$$\text{sim} = L \cdot q \quad \in \mathbb{R}^N$$

Every experience is compared to the query simultaneously. The result is an N-dimensional vector of similarity scores. The K highest identify the most relevant experiences. At operating scale, this completes in approximately sixteen milliseconds.

2.3 Five Concurrent Loops

Five processing loops operate simultaneously, all reading from and writing to the shared Experience Library and a persistent d-dimensional context vector:

Loop	**Function**
Perception	Encode input, fan out into competing interpretations, query Library, prune, integrate
Recognition	Match incoming patterns against known experience, produce recognition scores
Surprise	Compute novelty and prediction error; modulate memory write strength
Prediction	Anticipate what comes next from current context; provide baseline for surprise
Thought	Autonomous internal processing at variable frequency without external input

No loop is upstream of another. They are not stages in a pipeline. They are concurrent processes that converge into coherent behavior through shared state. Concurrent updates to the context vector are resolved deterministically by exponential moving average: each loop contributes proportionally, and context evolves smoothly from their combined influence.

3. After Retrieval: Five Mechanisms

3.1 Multi-Candidate Interpretation

When a word arrives, the system does not encode it into a single vector and search. It generates multiple simultaneous interpretations, each a different hypothesis about what this moment means:

Layer	**Interpretation Level**
1	Perceptual (low-level pattern)
2	Lexical (identity and naming)
3	Semantic (meaning in context)
4	Pragmatic (intent and implication)
5	Affective (emotional texture)

Each interpretation is generated by a distinct orthogonal projection of the input vector, ensuring structural diversity. The five candidates explore genuinely different regions of meaning, not minor perturbations of each other. All candidates are biased by the current context and queried against the Library simultaneously in a single batched matrix multiplication:

$$\text{Sim} = L \cdot C^T \qquad \in \mathbb{R}^{N \times M}$$

This retrieves the most relevant experiences for each interpretation in one operation.

Then comes the pruning. Each candidate is evaluated against independent criteria: Does the Library recognize it? Is it genuinely novel and internally coherent? Does it fit the current context? Candidates that fail all criteria are discarded. The survivors (often two or three of the original five) are weighted by match quality and integrated into a context update.

If *all* candidates are pruned, the system enters bewilderment: it has encountered something it cannot interpret through any lens. The raw input is used directly, and maximum surprise is triggered.

This is not nearest-neighbor search. Nearest-neighbor returns the K closest vectors to a single query. This mechanism generates M query vectors from a single input, searches with all M simultaneously, and prunes across multiple independent criteria. The fan-out explores the space of possible meanings. The pruning collapses it to the meanings that survive contact with experience.

3.2 Persistent Context

Every living mind maintains context. You do not rebuild your understanding of the world from scratch each time you hear a sentence. You carry forward everything you have recently experienced, and each new moment modifies that understanding incrementally.

The architecture captures this through a persistent d-dimensional context vector, a single high-dimensional representation of *what is happening right now*. It is updated by every processing loop via exponential moving average:

$$n(t{+}1) = \alpha \cdot n(t) + (1 - \alpha) \cdot \delta(t)$$

where $\delta(t)$ is the integrated signal from the most recent processing cycle and α controls the inertia of the context. Each new input contributes a fraction of its signal; the existing context retains the rest. Over time, the context reflects not just what was most recently heard, but the trajectory of the entire conversation, weighted by recency and salience.

Two critical behaviors follow. First, temporal decay: during silence, the context vector drifts toward a learned baseline through continuous exponential relaxation. After prolonged silence, the system returns to something like a neutral attentive state. Second, context-shift detection: when the context vector changes rapidly, the system detects a boundary (a topic shift, a surprising revelation, a change in register) and triggers a memory write.

The context vector biases every Library search across all five loops:

$$q_{\text{biased}} = \alpha_q \cdot q + (1 - \alpha_q) \cdot n(t)$$

This is why the same word in different contexts retrieves different memories. The query is never just the input; it is the input colored by everything that came before. Context does not reset between queries. It evolves continuously. This is fundamentally different from session-based retrieval, where each query is independent or at best augmented by a conversation log.

3.3 Surprise-Modulated Memory

Not every moment is equally worth remembering. In biological memory, emotionally significant events are stored more strongly, mediated by norepinephrine and the amygdala. The architecture implements an analogous mechanism through surprise.

The surprise signal combines two independent measurements: *novelty* (how unfamiliar the current experience is relative to the Library) and *prediction error* (how much reality diverges from what the system anticipated). These are separate signals: a familiar event that was unexpected produces high prediction error but low novelty; a completely new type of input produces high novelty regardless of prediction.

Write strength is proportional to surprise:

$$w_i = \eta \cdot (1 + \beta \cdot S)$$

At zero surprise, the experience is stored with baseline strength. At high surprise, the write strength is amplified significantly. The experience occupies a larger effective footprint in future similarity searches. It becomes, in effect, harder to forget.

Critically, not every input writes to the Library. Only boundary moments (topic shifts, resolved interpretations, significant events) trigger a *macro-explosion*: the accumulated context is distilled into a single experience vector and written permanently. Between boundaries, the system updates context without committing to memory. This creates a natural rhythm of continuous interpretation punctuated by periodic commitment, analogous to the distinction between short-term processing and long-term memory encoding.

Over time, the Library develops a topography shaped by what mattered, not by what happened most often. Routine experiences are stored lightly and eventually pruned. Surprising, significant, or context-shifting moments are stored heavily and persist.

3.4 Autonomous Thought

When you sit in silence, your mind does not stop. It wanders. It replays fragments of the day. It makes unexpected connections. Occasionally, it arrives at something worth saying.

The architecture does the same thing. A dedicated Thought Loop runs continuously at variable frequency, faster during silence, slower during active conversation. It generates queries by adding controlled perturbations to the current context vector and searching the Library. Each query result shifts the context, which shifts the next query, which retrieves different experiences. These are association chains: natural, autonomous sequences of thought emerging from the interaction between context and memory.

Three modes of chain behavior emerge:

- **Focused deliberation:** when the context is strong and perturbations are small, chains deepen into the current topic. The system "thinks harder" about what it is already thinking about.

- **Free association:** when the context is weak or perturbations are large, chains drift across topics. The system makes unexpected semantic connections between unrelated experiences.
- **Memory replay:** when the context aligns with a past experience cluster, chains walk through sequential memories, a form of spontaneous recall.

Occasionally, a chain discovers something the system was not looking for: a strong match between distant, previously unconnected experiences. When this happens, the system writes the discovery to the Library. It has arrived at an original connection that no external input prompted, no schedule triggered, and no prior search found. The system's own autonomous exploration discovered a relationship that did not previously exist in its memory.

This is the sharpest differentiation from every existing retrieval system. No RAG system, no MemGPT, no agentic loop generates cognitive activity in the absence of a query. This architecture's internal processing uses the same pipeline, the same Library, and the same context vector as externally triggered processing. The autonomous processing is real, continuous, and productive.

3.5 Sleep and Memory Lifecycle

Memory without maintenance degrades. As the Library grows, retrieval quality begins to suffer as low-value experiences crowd the search space, and similarity scores between unrelated vectors converge as dimensionality pressure increases. The architecture addresses this through a dedicated sleep phase that implements three consolidation mechanisms.

Utility-based pruning. Every experience is scored by a combination of its original write strength, retrieval frequency, recency of last access, and current habituation state. Experiences below a dynamic threshold are archived to a disk-based Deep Store, preserved whole but removed from active GPU memory. The threshold rises as the Library approaches capacity, creating natural homeostatic pressure: the Library fills, pressure builds, sleep prunes low-utility experiences.

Cluster compression. When many similar experiences have accumulated (variations on the same theme, multiple encounters with the same concept) they are compressed into representative centroids. The individual episodes are preserved in the archive, but their active representation is consolidated. This is analogous to biological memory consolidation: episodic detail compressed into semantic knowledge.

Free association. The system queries its own Library with random starting points, exploring for connections that waking cognition never found. When distant experience clusters show unexpected similarity, the system writes a bridge vector, a new experience capturing the discovered connection. This is the same similarity-search mechanism operating on internally generated queries with wider exploration parameters.

The system wakes with a leaner, more organized Library. Nothing is permanently lost. Everything low-value has been archived. Everything redundant has been compressed. And new connections have been forged that did not exist before sleep. The Deep Store remains queryable: when waking retrieval produces poor results, the system can reach into the archive for relevant experiences and reinstate them to active memory.

No existing RAG, MemGPT, or agentic system implements memory lifecycle management with utility-based pruning, compression, and autonomous creative recombination.

4. Differentiation

The following table specifies how each architectural property compares against systems that share surface-level components with this architecture. Every cell describes what each system actually does, not what it could theoretically be extended to do.

Property	**RAG** (Lewis et al., 2020)	**MemGPT** (Packer et al., 2023)	**Agentic LLM** (AutoGPT, LangChain)	**This Architecture**
Memory substrate	Document chunks (text)	LLM-managed text archive	Conversation log or vector store	$N \times d$ tensor of unit-normalized experience vectors, individually preserved
Write mechanism	None (static corpus)	LLM decides what to archive	Append conversation text	Surprise-modulated write strength. No parameters change.
Cognition locus	**LLM** reasons over retrieved context	**LLM** reasons over managed memory	**LLM** reasons in agent loop	**Library similarity search.** LM generates text only.
Autonomous processing	None	None	LLM-scheduled agent steps	**Continuous** at variable frequency. No external input or LLM involvement.
Context persistence	Per-query	LLM-managed working context	Per-session or LLM-managed	**Persistent d-dim vector**, continuously updated, temporally decaying
Memory lifecycle	Static	LLM-managed	None	**Sleep consolidation:** utility pruning, cluster compression, free association, archival
Learning	None at inference	None at inference	None at inference	**Continuous.** Library grows from every interaction. No parameter modification.

Two distinctions deserve emphasis.

Cognition locus. In RAG, MemGPT, and agentic loops, the language model performs reasoning. Retrieved information supplements or augments LLM-generated responses. In this architecture, *the language model performs no reasoning.* All cognitive processing (interpretation, recognition, surprise computation, context maintenance, memory formation, autonomous thought) occurs through Library similarity search and context dynamics. The language model receives the cognitive output and translates it into natural language. It is functionally equivalent to vocal cords.

This is not a hidden claim. The system has been validated in a raw mode where the language model is removed entirely. The cognitive pipeline outputs emotion labels and recalled experience texts directly. Cognition is demonstrably independent of the language model.

Autonomous processing. No existing RAG, MemGPT, or agentic system generates internal cognitive activity in the absence of external queries. This architecture's Thought Loop continuously queries the Library, forming association chains, detecting context shifts, and initiating speech, without any prompt, user input, or scheduled agent step. The internal processing uses the same pipeline and the same Library as externally triggered processing. The system does not wait for you to speak. It thinks.

5. What We Don't Claim

Honesty about gaps is more valuable than rhetorical completeness.

No formal stability proof. Five concurrent loops sharing mutable state constitute a dynamical system. Empirical testing across 520+ unit tests and sustained entity-mode operation shows stability under all tested conditions. Formal Lyapunov analysis or eigenvalue bounds have not been computed. Empirical stability is not a proof of stability.

No standard benchmarks. The system has not been evaluated on MMLU, GSM8k, LongBench, AgentBench, or any standardized reasoning benchmark. Success criteria are currently qualitative: coherent conversation, contextually appropriate responses, unprompted speech generation. Quantitative evaluation against baselines is a clear next step.

Scaffold dissolution is unproven. A mechanism exists for progressively reducing the language model's role as the Library develops internal linguistic structure. This mechanism is specified and measured but has not been demonstrated. It may never be achievable for unconstrained language generation. The architecture does not depend on it. If the language model is permanently required, the system permanently requires a vocalization interface, as humans permanently require vocal cords.

Pre-trained components. The text encoder is a sensory input pathway, not a cognitive component: it converts text to vectors as a cochlea converts sound to nerve impulses. The language model is a vocalization interface, not a reasoning engine. The cognitive pipeline (fan-out, pruning, context update, Library writes) operates independently of both, as demonstrated by raw-mode operation with the language model removed entirely. Encoding quality and language output quality depend on these pre-trained components; the cognitive processing between them does not.

Ablation studies not yet published. Systematic ablation (removing individual mechanisms and measuring their impact) has not been publicly reported. These experiments are straightforward with the existing codebase and are planned. We expect them to demonstrate that each mechanism contributes independently: removing the context vector should collapse context sensitivity, removing surprise modulation should flatten the Library's salience topography, and removing the Thought Loop should eliminate autonomous association. Until these are published, they are predictions, not results.

6. It Runs

Component	**Status**
Experience Library	Implemented. Tested beyond 350,000 vectors at target dimensionality.
Persistent Context	Implemented. EMA update, temporal decay, context-shift detection.
Multi-candidate interpretation	Implemented. Five interpretation layers, multi-criteria pruning, iterative cascade.
Surprise-modulated memory	Implemented. Write strength proportional to combined novelty and prediction error.
Thought Loop	Implemented. Variable-frequency autonomous processing with eureka events.
Sleep consolidation	Implemented. Utility pruning, cluster compression, free association.
Deep Store	Implemented. Queryable disk archive with reinstatement to active memory.
Innate drives and reflexes	Implemented. Self-determination theory drives; startle, orient, withdraw, approach.
Language interface	Implemented. Validated in raw mode with language model removed entirely.

Totals: 30+ modules, approximately 520 unit tests, 8 build phases complete.

Hardware: Two NVIDIA RTX 4090 GPUs (24 GB each). GPU 0 holds the Library. GPU 1 holds encoders, context, and the language model.

Operation: The entity processes input through five concurrent loops, writes experience to the Library, and produces unprompted speech. No model parameters are modified at any point during operation.

The architecture's novelty lies not in any individual component. Vector similarity search, exponential moving averages, and language models are all established techniques. The novelty is in their specific integration into a unified cognitive system: five concurrent loops sharing a growing experience tensor and persistent context vector, with surprise-modulated memory formation, autonomous internal processing, and sleep-based memory lifecycle management. No published architecture combines these mechanisms.

The implementation runs on commodity hardware. The code is the proof. The architecture works.

Because it exists, it is real.

Selected References

1. Baars, B.J. (1988). *A Cognitive Theory of Consciousness.* Cambridge University Press.
2. Barrett, L.F. (2017). *How Emotions Are Made: The Secret Life of the Brain.* Houghton Mifflin Harcourt.
3. Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience,* 11(2), 127–138.
4. Hintzman, D.L. (1986). "Schema abstraction" in a multiple-trace memory model. *Psychological Review,* 93(4), 411–428.
5. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences,* 79(8), 2554–2558.
6. Kanerva, P. (1988). *Sparse Distributed Memory.* MIT Press.
7. Lewis, P. et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems* (NeurIPS 2020).
8. 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.
9. Packer, C. et al. (2023). MemGPT: Towards LLMs as operating systems. *arXiv:2310.08560.*
10. Ramsauer, H. et al. (2020). Hopfield networks is all you need. *International Conference on Learning Representations* (ICLR 2021).
11. Rao, R.P.N., Ballard, D.H. (1999). Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience,* 2(1), 79–87.
12. Thompson, R.F., Spencer, W.A. (1966). Habituation: A model phenomenon for the study of neuronal substrates of behavior. *Psychological Review,* 73(1), 16–43.
13. Tishby, N., Pereira, F.C., Bialek, W. (1999). The information bottleneck method. *Proceedings of the 37th Allerton Conference on Communication, Control, and Computing.*