Geometric Mnemic Manifolds: A Position Paper

Toward Structured Memory Architectures for Persistent AI Agency

With Formal Complexity Analysis and Research Agenda

Alan Garcia
Independent Researcher
alan.javier.garcia@gmail.com

December 8, 2025
Version 3.0 (Position Paper)

Abstract

Status: Theoretical Specification. This paper presents the **Geometric Mnemic Manifold (GMM)**, a proposed memory architecture for large language models that externalizes the Key-Value cache to a distributed store with geometrically enforced sparse attention. We provide formal complexity proofs for coordinate addressing ($\mathcal{O}(1)$), hierarchical retrieval ($\mathcal{O}(\log N)$), and active edge bounds. Crucially, we distinguish proven properties from conjectured benefits, identify critical validation gates, and frame GMM as a *research agenda* rather than a proven solution. The core innovations—entropy-gated reification, polynomial temporal decay, and Kronecker-sequence addressing—are presented with mathematical rigor where possible and honest acknowledgment of open problems where not. We argue that the fundamental question facing AI memory systems is not capacity but *structure*: whether raw context windows suffice for deployed agency or whether architectural organization is necessary for auditability, compositionality, and long-horizon coherence.

Contents

1 Introduction and Motivation **3**
1.1 The Context Window Arms Race . . . 3
1.2 Thesis Statement . . . 3
1.3 Scope and Epistemic Status . . . 3

2 Related Work **3**
2.1 Memory-Augmented Neural Networks . . . 3
2.2 Retrieval-Augmented Generation . . . 4
2.3 Long-Context Transformers . . . 4
2.4 Agent Memory Systems . . . 4
2.5 Discrepancy Theory . . . 4

3 Formal Framework **4**
3.1 Notation . . . 4
3.2 The Engram and the Manifold . . . 5
3.3 Kronecker Sequence Addressing . . . 5
3.4 Low-Discrepancy Coverage . . . 6
3.5 Hierarchical Layer Structure . . . 6
3.6 Retrieval Complexity . . . 7

4 Entropy-Gated Reification **8**
4.1 Attention Entropy . . . 8
4.2 The Reification Criterion . . . 9

5 The Recursive Reasoning Kernel **9**
5.1 Definition and Role . . . 9
5.2 Epistemic Regularization . . . 9

6 Value Propositions **10**
6.1 Geometric Auditability . . . 10
6.2 Compositional Potential . . . 10
6.3 Index-Free Instantiation . . . 10
6.4 Configurable Temporal Bias . . . 10

7 Limitations and Anti-Patterns **10**
7.1 Storage Overhead . . . 11
7.2 Operational Complexity . . . 11
7.3 Anti-Patterns . . . 11

8 Validation Roadmap **11**
8.1 Phase 0: Epistemic Gap Detection (3–6 months) . . . 11
8.2 Phase 1: Synthetic Benchmarks (2–3 months) . . . 11
8.3 Phase 2: Domain Deployment (6–12 months) . . . 12
8.4 Phase 3: Multi-Agent Composition (6–12 months) . . . 12

9 Open Questions **12**

10 Conclusion **12**
10.1 Summary of Contributions . . . 12
10.2 The Fundamental Question . . . 13
10.3 Call to Action . . . 13

A Comparison Matrix **14**

B Cost Model Example **14**

C Notation Summary **14**

1 Introduction and Motivation

1.1 The Context Window Arms Race

Current foundation models pursue memory through architectural brute force: extending token windows from 4K to 128K to 1M tokens and beyond. This approach implicitly assumes that *capacity* alone solves the memory problem. We argue this assumption is flawed for deployed, long-lived AI agents.

(i) **Quadratic Attention Cost**: Standard self-attention computes $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$, requiring $\mathcal{O}(N^2)$ operations in sequence length N. Despite advances in sparse attention [1, 3], ultra-long contexts remain computationally prohibitive.

(ii) **Black Box Explainability**: When a 10M-token context produces an error, identifying the causative information becomes archaeologically complex. There is no principled way to audit which memories influenced a decision.

(iii) **Architectural Rigidity**: Merging knowledge from multiple specialized agents requires either full retraining or fragile prompt engineering. Context windows do not compose.

1.2 Thesis Statement

> *The Geometric Mnemic Manifold proposes that persistent AI agency requires **structured memory**—an explicit separation of fluid reasoning (the kernel) from crystallized knowledge (the manifold)—with geometric organization enabling formal guarantees on retrieval complexity, auditability, and compositionality.*

1.3 Scope and Epistemic Status

This is a **position paper** presenting a research agenda. We distinguish:

- **Proven**: Mathematical properties of the geometric construction (Section 3)
- **Conjectured**: Benefits for AI systems (auditability, compositionality)
- **Unvalidated**: Whether the architecture works in practice (requires empirical testing per Section 8)

We have not built a production system. The value of this paper lies in (a) formalizing the mathematical framework, (b) identifying critical validation gates, and (c) stimulating research in structured memory for AI.

2 Related Work

GMM builds on and differs from several lines of research.

2.1 Memory-Augmented Neural Networks

Neural Turing Machines (NTM) [4] and **Differentiable Neural Computers (DNC)** [5] introduced external memory banks with learned read/write heads. GMM differs in using *geometric* rather than *learned* addressing, trading flexibility for formal guarantees.

2.2 Retrieval-Augmented Generation

RAG systems [7] retrieve relevant documents via similarity search before generation. **RETRO** [2] scales this to trillions of tokens with chunked cross-attention. GMM differs by:

- Encoding temporal structure geometrically (not as metadata)
- Hierarchical abstraction (L0/L1/L2) rather than flat retrieval
- Deterministic addressing rather than approximate nearest neighbor

2.3 Long-Context Transformers

Longformer [1] and **BigBird** [13] use sparse attention patterns to achieve linear scaling. GMM's geometric sparsity is *enforced by topology* rather than learned, providing formal guarantees at the cost of flexibility.

2.4 Agent Memory Systems

Generative Agents [10] maintain memory streams with recency-importance-relevance scoring. **MemGPT** [9] introduces tiered memory management. GMM differs by:

- Geometric encoding of time (radial coordinate) vs. metadata tags
- Formal complexity bounds vs. heuristic scoring
- Hierarchical semantic compression vs. flat storage

2.5 Discrepancy Theory

GMM's use of Kronecker sequences derives from **quasi-Monte Carlo** methods [8]. The low-discrepancy property ensures uniform coverage of the hypersphere, which we exploit for memory addressing.

3 Formal Framework

We now present the mathematical foundations of GMM with formal definitions, theorems, and proofs.

3.1 Notation

Notation 3.1. Throughout this paper:

- $N \in \mathbb{N}$ denotes the total number of engrams (memory units)
- $d \in \mathbb{N}$ denotes embedding dimension (typically 768–4096)
- $\mathbb{S}^{d-1} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| = 1\}$ is the unit hypersphere
- $\{x\} = x - \lfloor x \rfloor$ denotes the fractional part
- $\alpha = (\sqrt{p_1}, \sqrt{p_2}, \ldots, \sqrt{p_{d-1}})$ for distinct primes p_i

3.2 The Engram and the Manifold

Definition 3.2 (Engram)**.** An **engram** is a tuple $\varepsilon_k = (k, W_k, \mathbf{e}_k, t_k, H_k)$ where:

- $k \in \mathbb{N}$ is the temporal index
- W_k is the serialized context window (tokens) at time t_k
- $\mathbf{e}_k \in \mathbb{R}^d$ is the embedding vector
- $t_k \in \mathbb{R}_{\geq 0}$ is the timestamp
- $H_k \in \mathbb{R}_{\geq 0}$ is the attention entropy at creation (Definition 4.1)

Definition 3.3 (Geometric Mnemic Manifold)**.** A **Geometric Mnemic Manifold** is a triple $\mathcal{M} = (\mathcal{E}, \phi, \mathcal{L})$ where:

- $\mathcal{E} = \{\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{N-1}\}$ is the set of engrams
- $\phi : \mathbb{N} \to \mathbb{S}^{d-1} \times \mathbb{R}_{>0}$ is the coordinate function (Definition 3.4)
- $\mathcal{L} = (L_0, L_1, L_2)$ is the hierarchical layer structure (Definition 3.11)

3.3 Kronecker Sequence Addressing

Definition 3.4 (Coordinate Function)**.** The coordinate function $\phi : \mathbb{N} \to \mathbb{S}^{d-1} \times \mathbb{R}_{>0}$ maps temporal index k to position $(\mathbf{u}_k, r_k)$ via:

$$\mathbf{u}_k = \mathcal{N}\left(\text{erf}^{-1}(2\{k\alpha_1\} - 1), \ldots, \text{erf}^{-1}(2\{k\alpha_{d-1}\} - 1)\right) \tag{1}$$

$$r_k = (1 + k)^{-\gamma} \quad \text{for } \gamma > 0 \tag{2}$$

where $\mathcal{N} : \mathbb{R}^{d-1} \to \mathbb{S}^{d-1}$ normalizes to the unit sphere and $\alpha_i = \sqrt{p_i}$ for the i-th prime.

Remark 3.5 (Choice of Radial Function). We use polynomial decay $(1 + k)^{-\gamma}$ rather than exponential decay $e^{-\lambda k}$ to preserve heavy-tailed retrieval probability. At $k = 10^6$ with $\lambda = 0.015$: $e^{-\lambda k} \approx e^{-15000} \approx 0$, while $(1 + k)^{-1} \approx 10^{-6}$. The polynomial form ensures ancient but highly relevant memories retain non-negligible accessibility.

Theorem 3.6 (O(1) Address Calculation)**.** *For any temporal index $k \in \mathbb{N}$, the coordinate $\phi(k) = (\mathbf{u}_k, r_k)$ can be computed in $\mathcal{O}(d)$ time, which is $\mathcal{O}(1)$ for fixed embedding dimension d.*

Proof. The computation of $\phi(k)$ requires:

1. Computing $\{k\alpha_i\}$ for $i = 1, \ldots, d - 1$: Each fractional part requires one multiplication and one floor operation, giving $\mathcal{O}(d - 1) = \mathcal{O}(d)$ operations.
2. Computing $\text{erf}^{-1}(2\{k\alpha_i\} - 1)$: The inverse error function can be computed to machine precision via rational approximations in $\mathcal{O}(1)$ per coordinate [12], giving $\mathcal{O}(d)$ total.
3. Normalization $\mathcal{N}$: Computing the norm requires $\mathcal{O}(d)$ operations (sum of squares, square root) and dividing each component requires $\mathcal{O}(d)$.
4. Radial coordinate: $(1 + k)^{-\gamma}$ requires $\mathcal{O}(1)$ operations.

Total: $\mathcal{O}(d)$. For fixed d (a constant in typical applications where $d \in \{768, 1024, 1536, 4096\}$), this is $\mathcal{O}(1)$. □

Remark 3.7 (Address vs. Retrieval). Theorem 3.6 proves that *computing coordinates* is $\mathcal{O}(1)$. This does **not** imply that *retrieval* is $\mathcal{O}(1)$. Retrieval requires:

1. Address calculation: $\mathcal{O}(1)$ (proven)
2. Storage lookup: $\mathcal{O}(1)$ with hash tables, $\mathcal{O}(\log N)$ with B-trees
3. Semantic matching: $\mathcal{O}(k)$ for comparing query to k candidates

The full retrieval complexity depends on implementation and is analyzed in Section 3.6.

3.4 Low-Discrepancy Coverage

Definition 3.8 (Star Discrepancy)**.** For a point set $P = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset [0,1)^d$, the **star discrepancy** is:

$$D_N^*(P) = \sup_{\mathbf{y} \in [0,1]^d} \left| \frac{|\{i : \mathbf{x}_i \in [0, \mathbf{y})\}|}{N} - \prod_{j=1}^{d} y_j \right| \tag{3}$$

where $[0, \mathbf{y}) = [0, y_1) \times \cdots \times [0, y_d)$.

Theorem 3.9 (Kronecker Sequence Discrepancy [6, Theorem 3.6])**.** *Let $\alpha = (\alpha_1, \ldots, \alpha_d)$ where $1, \alpha_1, \ldots, \alpha_d$ are linearly independent over . The sequence $\mathbf{x}_k = \{k\alpha\}$ satisfies:*

$$D_N^*(\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}) = \mathcal{O}\left(\frac{(\log N)^d}{N}\right) \tag{4}$$

Corollary 3.10 (Uniform Angular Coverage)**.** *The angular components $\mathbf{u}_k$ of GMM coordinates are equidistributed on $\mathbb{S}^{d-1}$ in the sense that for any measurable $A \subseteq \mathbb{S}^{d-1}$:*

$$\lim_{N \to \infty} \frac{|\{k \leq N : \mathbf{u}_k \in A\}|}{N} = \frac{vol(A)}{vol(\mathbb{S}^{d-1})} \tag{5}$$

Proof. The map $\mathbf{z} \mapsto \mathcal{N}(\text{erf}^{-1}(2z_1 - 1), \ldots, \text{erf}^{-1}(2z_{d-1} - 1))$ is measure-preserving from $[0,1)^{d-1}$ with Lebesgue measure to $\mathbb{S}^{d-1}$ with uniform measure. By Weyl's equidistribution theorem and Theorem 3.9, the sequence $\{k\alpha\}$ is equidistributed in $[0,1)^{d-1}$, hence the transformed sequence is equidistributed on $\mathbb{S}^{d-1}$. □

3.5 Hierarchical Layer Structure

Definition 3.11 (Layer Structure)**.** Given parameters $\beta_1, \beta_2 \in \mathbb{N}$ (default: $\beta_1 = 64$, $\beta_2 = 16$), define:

$$L_0 = \mathcal{E} \quad \text{(raw episodic engrams)} \tag{6}$$

$$|L_1| = \left\lceil \frac{N}{\beta_1} \right\rceil \quad \text{(synthesized pattern nodes)} \tag{7}$$

$$|L_2| = \left\lceil \frac{N}{\beta_1 \cdot \beta_2} \right\rceil \quad \text{(semantic axiom nodes)} \tag{8}$$

Each L_1 node summarizes β_1 consecutive L_0 engrams via a telegraphic compressor Λ. Each L_2 node abstracts β_2 consecutive L_1 nodes.

Definition 3.12 (Foveated Connectivity)**.** Let $k_{\text{fovea}} \in \mathbb{N}$ (default: 10) and $k_{\text{para}} = \beta_1$. The active agent at position 0 maintains connections:

- **Fovea**: Direct edges to all ε_k with $k < k_{\text{fovea}}$
- **Para-fovea**: Direct edges to all L_1 nodes covering $k \in [k_{\text{fovea}}, k_{\text{para}})$
- **Periphery**: Direct edges to all L_2 nodes covering $k \geq k_{\text{para}}$

Theorem 3.13 (Active Edge Bound)**.** *Under the foveated connectivity of Definition 3.12, the number of active edges $E(N)$ satisfies:*

$$E(N) = k_{fovea} + \left\lceil \frac{k_{para} - k_{fovea}}{\beta_1} \right\rceil + \left\lceil \frac{N - k_{para}}{\beta_1 \cdot \beta_2} \right\rceil \tag{9}$$

Proof. By Definition 3.12:

1. Foveal edges: exactly k_{fovea} (connecting to raw engrams $k < k_{\text{fovea}}$)
2. Para-foveal edges: the para-foveal region spans indices $[k_{\text{fovea}}, k_{\text{para}})$, covered by $\lceil (k_{\text{para}} - k_{\text{fovea}})/\beta_1 \rceil$ L_1 nodes
3. Peripheral edges: indices $[k_{\text{para}}, N)$ are covered by $\lceil (N - k_{\text{para}})/(\beta_1 \cdot \beta_2) \rceil$ L_2 nodes

Summing gives the stated formula. □

Corollary 3.14 (Linear Growth with Small Constant)**.** *For the default parameters ($k_{fovea} = 10$, $k_{para} = 64$, $\beta_1 = 64$, $\beta_2 = 16$):*

$$E(N) = 10 + 1 + \left\lceil \frac{N - 64}{1024} \right\rceil \approx 11 + \frac{N}{1024} \tag{10}$$

This is $\mathcal{O}(N)$ growth, but with a factor of $\approx 1/1000$ improvement over naive $\mathcal{O}(N)$.

Remark 3.15 (Honest Complexity Assessment). Earlier versions of this work claimed $\mathcal{O}(1)$ active edges. This is **incorrect**. Corollary 3.14 shows linear growth. The improvement over naive storage is a constant factor ($\approx 1000\times$), which is significant in practice but does not change asymptotic complexity.

True $\mathcal{O}(\log N)$ active edges would require recursive hierarchical layers (L3, L4, ...) with logarithmic depth. This is an open design question (Section 9).

3.6 Retrieval Complexity

Definition 3.16 (Dual-Metric Retrieval)**.** Given query embedding $\mathbf{q} \in \mathbb{R}^d$ and engram ε_k with embedding $\mathbf{e}_k$, the **retrievability score** is:

$$R(\mathbf{q}, \varepsilon_k) = \text{sim}(\mathbf{q}, \mathbf{e}_k) \cdot r_k^{\rho} \tag{11}$$

where $\text{sim}(\cdot, \cdot)$ is cosine similarity and $\rho > 0$ controls temporal weighting.

Theorem 3.17 (Hierarchical Retrieval Complexity)**.** *The GMM retrieval algorithm (Algorithm 1) runs in:*

$$T(N) = \mathcal{O}\left(k_{fovea} + |L_1| + \beta_1\right) = \mathcal{O}\left(k_{fovea} + \frac{N}{\beta_1} + \beta_1\right) \tag{12}$$

For fixed k_{fovea} and β_1, this is $\mathcal{O}(N/\beta_1) = \mathcal{O}(N)$ but with a $1/64$ constant factor reduction.

Proof. Algorithm 1 performs:

1. Foveal scan: Compare query to k_{fovea} engrams: $\mathcal{O}(k_{\text{fovea}} \cdot d)$
2. Pattern broadcast: Compare query to $|L_1| = \lceil N/\beta_1 \rceil$ summary nodes: $\mathcal{O}(|L_1| \cdot d)$
3. Drill-down: If a L_1 node matches, examine its β_1 children: $\mathcal{O}(\beta_1 \cdot d)$

Total: $\mathcal{O}((k_{\text{fovea}} + |L_1| + \beta_1) \cdot d)$. For fixed d, this reduces to the stated bound. □

Algorithm 1 Hierarchical GMM Retrieval

Require: Query $\mathbf{q}$, manifold $\mathcal{M}$, threshold τ
Ensure: Top-k matching engrams

candidates $\leftarrow \emptyset$ ▷ Phase 1: Foveal scan
for $i = 0$ to $k_{\text{fovea}} - 1$ **do**
  if $R(\mathbf{q}, \varepsilon_i) > \tau$ **then**
    candidates $\leftarrow$ candidates $\cup \{\varepsilon_i\}$
  end if
end for ▷ Phase 2: Pattern broadcast
for each $\ell \in L_1$ **do**
  if $\text{sim}(\mathbf{q}, \mathbf{e}_\ell) > \tau$ **then** ▷ Phase 3: Drill-down
    for each child ε_j of ℓ **do**
      if $R(\mathbf{q}, \varepsilon_j) > \tau$ **then**
        candidates $\leftarrow$ candidates $\cup \{\varepsilon_j\}$
      end if
    end for
  end if
end for
return top-k from candidates by $R(\mathbf{q}, \cdot)$

Remark 3.18 (Achieving True O(log N) Retrieval). To achieve $\mathcal{O}(\log N)$ retrieval, one would need:

1. **Recursive hierarchy**: $L_3, L_4, \ldots$ with depth $\mathcal{O}(\log N)$
2. **Constant branching factor**: Each node has $\mathcal{O}(1)$ children
3. **Learned routing**: A classifier to navigate the hierarchy without exhaustive scan

This is an open research direction (Open Question 1).

4 Entropy-Gated Reification

A critical question for any external memory system is: which states warrant full computational instantiation? We propose an information-theoretic criterion.

4.1 Attention Entropy

Definition 4.1 (Attention Entropy)**.** Let $A_t \in \mathbb{R}^{n \times n}$ be the attention matrix at time t, with rows $\mathbf{a}_i = \text{softmax}(\mathbf{q}_i^\top K / \sqrt{d_k})$. The **attention entropy** is:

$$\text{H}(A_t) = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \log a_{ij} \tag{13}$$

Remark 4.2 (Interpretation). High entropy ($\text{H} \to \log n$) indicates diffuse attention—the model is "uncertain" and attending broadly. Low entropy ($\text{H} \to 0$) indicates focused attention on few tokens. We hypothesize that high-entropy states represent cognitively complex moments warranting preservation.

4.2 The Reification Criterion

Definition 4.3 (Reification Function)**.** A stored state S_t is **reified** (promoted to active computational instance) if:

$$\text{Reify}(t) = \begin{cases} \text{True} & \text{if } (t_{\text{now}} - t) < \tau_{\text{recency}}, \text{ or} \\ \text{True} & \text{if } \text{H}(A_t) > \theta_{\text{threshold}} \\ \text{False} & \text{otherwise} \end{cases} \tag{14}$$

Remark 4.4 (Computational Cost of Entropy). Computing $\text{H}(A_t)$ exactly requires access to the full attention matrix, which has $\mathcal{O}(n^2)$ entries. This creates a circularity: we must compute expensive attention to decide whether to *store* the state.

Proposed mitigation: Use a proxy measure:

1. **Output perplexity**: High perplexity in generated tokens correlates with attention uncertainty
2. **Foveal entropy**: Compute entropy over only the k_{fovea} most recent attention rows
3. **Learned classifier**: Train a small network to predict "interestingness" from hidden states

Empirical validation of these proxies is required (Section 8, Phase 0).

5 The Recursive Reasoning Kernel

5.1 Definition and Role

Definition 5.1 (Recursive Reasoning Kernel)**.** The **RRK** is a language model $\mathcal{M}_\theta$ optimized for:

1. **Fluid intelligence**: Reasoning, synthesis, and planning
2. **Epistemic gap detection**: Recognizing when required information is absent from context
3. **Bus signaling**: Emitting a special token `<SIGNAL_BUS>` to trigger manifold retrieval

Candidate architectures: Phi-3 Mini (3.8B), Qwen 2.5 (0.5B–7B).

5.2 Epistemic Regularization

Definition 5.2 (Signal Loss)**.** Let $\mathcal{D}_{\text{gap}}$ be a distribution over (q, W, m) where q is a query, W is (possibly masked) context, and $m \in \{0, 1\}$ indicates whether W contains sufficient information. Define:

$$\mathcal{L}_{\text{signal}} = -\mathbb{E}_{(q,W,m)\sim\mathcal{D}_{\text{gap}}} \left[(1-m)\log p_\theta(s|q,W) + m\log(1 - p_\theta(s|q,W))\right] \tag{15}$$

where $s =$ `<SIGNAL_BUS>` and $p_\theta(s|q,W)$ is the probability the RRK emits s as first response token.

Definition 5.3 (Total Training Loss)**.** The RRK is trained on:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{LM}} + \lambda\mathcal{L}_{\text{signal}} \tag{16}$$

where $\mathcal{L}_{\text{LM}}$ is standard language modeling loss and $\lambda \in [0.3, 0.7]$ balances the objectives.

Remark 5.4 (Critical Validation Gate). The entire GMM architecture depends on the RRK reliably detecting epistemic gaps. If small models cannot learn this with $\geq 90\%$ precision and recall, GMM collapses to standard RAG with extra complexity. This is the **make-or-break** validation (Phase 0, Section 8).

6 Value Propositions

We claim GMM offers advantages over standard approaches. These are **conjectures** requiring empirical validation.

6.1 Geometric Auditability

Claim: Because engram positions are deterministic functions of temporal index, retrieval paths are reproducible.

Mechanism: Same query $\mathbf{q}$ at same manifold state $\rightarrow$ same Kronecker coordinates $\rightarrow$ identical candidate engrams.

Limitation: This provides auditability of *geometric access patterns*, not semantic encoding. Why the embedding model places $\mathbf{q}$ near certain engrams remains opaque.

Value: Significant improvement over stochastic indices (HNSW) where rebuilding subtly changes neighbors. Essential for regulated industries.

6.2 Compositional Potential

Claim: Multiple manifolds can be mounted to a single RRK because they share a universal address space (the Kronecker spiral).

Requirements:

1. **Embedding alignment**: All agents must use identical embedding models
2. **Semantic drift mitigation**: Manifolds from different time periods may encode incompatible semantics

Value: Enables asynchronous knowledge scaling without centralized retraining.

6.3 Index-Free Instantiation

Claim: GMM eliminates index loading overhead because the "index" is an equation (Theorem 3.6).

Reality check: Agent startup still requires:

- Loading RRK weights: 1–2GB, standard for small LMs
- Streaming engrams from storage: Network I/O, implementation-dependent

Value: Eliminates compounding latency from graph traversal. Enables serverless agent architectures.

6.4 Configurable Temporal Bias

Claim: The radial decay function encodes a "forgetting curve" as intrinsic topology rather than extrinsic filtering.

Tuning: Parameter γ in $r_k = (1 + k)^{-\gamma}$ controls decay:

- Customer service: $\gamma = 2$ (aggressive decay, recent policies)
- Legal research: $\gamma = 0.5$ (slow decay, centuries of precedent)

7 Limitations and Anti-Patterns

GMM is **not** appropriate for all applications.

7.1 Storage Overhead

Cost: Every engram is persisted. Storage grows $\mathcal{O}(N)$ with experience, unlike weight-based models where knowledge is compressed into fixed parameters.

Estimate: 10^6 engrams $\times$ 4KB average $\approx$ 4GB storage.

7.2 Operational Complexity

GMM requires managing: embedding service, storage backend, cache layer, RRK deployment, manifold versioning. This is significantly more complex than stateless LLM inference.

7.3 Anti-Patterns

1. **High-frequency, low-stakes**: FAQ bots don't need persistent memory
2. **Ultra-low latency**: 50ms retrieval exceeds robotics/HFT budgets
3. **Privacy-first contexts**: Total retention conflicts with "right to be forgotten"
4. **Rapidly evolving domains**: Immutable engrams become stale baggage

8 Validation Roadmap

We propose a phased validation strategy. Each phase has explicit success criteria and failure modes.

8.1 Phase 0: Epistemic Gap Detection (3–6 months)

Goal: Prove small models can reliably learn when to signal retrieval.

Method:

- Train RRK (Qwen 2.5 0.5B or Phi-3 Mini 3.8B) on masked context dataset
- 50% samples: full context (should NOT signal)
- 50% samples: answer-relevant sentences masked (SHOULD signal)

Success criteria:

- Precision $\geq 90\%$ (doesn't hallucinate when info present)
- Recall $\geq 90\%$ (signals when info missing)
- Stable training with $\lambda \in [0.3, 0.7]$

Failure mode: If small models cannot learn this, GMM offers no advantage over standard RAG.

8.2 Phase 1: Synthetic Benchmarks (2–3 months)

Goal: Empirically measure retrieval speedup vs. HNSW.

Method: "Needle in the Spiral" benchmark

- Insert passkey at random depth k in manifolds of size $10^3, 10^4, 10^5, 10^6$
- Measure Time-to-First-Token (TTFT)
- Compare GMM vs. HNSW baseline

Success criteria:

- GMM TTFT grows slower than HNSW with N
- Recall@1 $\geq 95\%$

8.3 Phase 2: Domain Deployment (6–12 months)

Goal: Demonstrate auditability value in high-stakes domain.

Method: Pilot in legal document analysis or medical diagnosis support. Instrument retrieval paths for compliance audits.

Success criteria:

- 10× reduction in audit preparation time
- Zero "unexplainable retrieval" incidents

8.4 Phase 3: Multi-Agent Composition (6–12 months)

Goal: Validate manifold merging without catastrophic interference.

Method: Train Agent A (contract law) and Agent B (patent law). Mount both to unified RRK. Test on cross-domain queries.

Success criteria:

- Composed accuracy $\geq 90\%$ of specialist agents
- Correct provenance attribution

9 Open Questions

Open Question 1 (True O(log N) Retrieval). Can hierarchical GMM achieve $\mathcal{O}(\log N)$ retrieval via recursive layers (L3, L4, ...) with learned routing? What are the trade-offs between depth, branching factor, and accuracy?

Open Question 2 (Optimal Decay Functions). Is polynomial decay $(1+k)^{-\gamma}$ universally optimal? How do exponential, hyperbolic, and piecewise functions compare across domains?

Open Question 3 (Entropy Proxies). What cheap proxy best approximates attention entropy for reification gating? Does output perplexity suffice, or is a learned classifier necessary?

Open Question 4 (Multi-Modal Engrams). How can GMM extend to images, audio, and video? What embedding spaces preserve geometric structure across modalities?

Open Question 5 (Adversarial Robustness). Can attackers inject "poison engrams" to bias retrieval? What defenses exist beyond cryptographic signing?

10 Conclusion

10.1 Summary of Contributions

This position paper contributes:

1. **Formal framework**: Rigorous definitions and complexity proofs for GMM (Section 3)
2. **Honest assessment**: Correction of overclaimed complexity bounds (Corollary 3.14)
3. **Validation roadmap**: Concrete phases with success criteria and failure modes (Section 8)
4. **Research agenda**: Open questions for the community (Section 9)

10.2 The Fundamental Question

Context windows will reach 10 million tokens—they already are. The question is whether **raw capacity** suffices for deployed agency, or whether we need **architectural structure** for auditability, compositionality, and long-horizon coherence.

We argue the latter. Just as databases impose structure (tables, indices, constraints) on raw bytes for reliability and query efficiency, GMM proposes geometric structure on semantic memory for the same reasons.

10.3 Call to Action

This is a research agenda, not a solved problem. We invite:

- **Empiricists**: Validate Phase 0 (epistemic gap detection) as the critical gate
- **Theorists**: Achieve true $\mathcal{O}(\log N)$ retrieval via recursive hierarchy
- **Practitioners**: Identify domains where auditability justifies architectural complexity

The future of AI memory may not be infinite context—it may be *structured memory*.

References

[1] Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

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

[3] Child, R., Gray, S., Radford, A., and Sutskever, I. (2019). Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

[4] Graves, A., Wayne, G., and Danihelka, I. (2014). Neural Turing machines. *arXiv preprint arXiv:1410.5401*.

[5] Graves, A., et al. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476.

[6] Kuipers, L. and Niederreiter, H. (1974). *Uniform Distribution of Sequences*. Wiley-Interscience.

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

[8] Niederreiter, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM.

[9] Packer, C., Fang, V., Patil, S. G., Lin, K., Wooders, S., and Gonzalez, J. E. (2023). MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*.

[10] Park, J. S., et al. (2023). Generative agents: Interactive simulacra of human behavior. In *UIST*.

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

[12] Winitzki, S. (2008). A handy approximation for the error function and its inverse. Lecture note, available online.

[13] Zaheer, M., et al. (2020). Big Bird: Transformers for longer sequences. In *NeurIPS*.

A Comparison Matrix

Feature	Standard LLM	Vector DB (RAG)	GMM
Learning	Gradient descent	Append + index	Append to spiral
Auditability	Low (black box)	Medium (stochastic)	High (geometric)[†]
Unlearning	Retrain (expensive)	Re-index ($\mathcal{O}(N \log N)$)	Excise node[‡]
Composition	Impossible	Hard (merge indices)	Possible[§]
Cold Start	Fast	Slow (load index)	Medium[‖]
Storage	Low (compressed)	Medium (vectors)	High (all engrams)
Temporal Reasoning	None	Metadata tag	Geometric coordinate

Table 1: Comparison of memory architectures. [†]Geometric position auditable; embedding logic remains opaque. [‡]Requires handling coherence gaps. [§]Requires shared embedding models. [‖]Index-free but requires RRK loading.

B Cost Model Example

Scenario: Legal research assistant with 5 years of operation.

- **Engrams**: $\sim 10^6$ interactions $\rightarrow$ 50GB storage (S3: \$1.15/month)
- **RRK**: Phi-3 Mini 3.8B $\rightarrow$ 7.6GB weights $\rightarrow$ 0.5s load time
- **Cache**: 100MB Redis (\$10/month)
- **Compute**: 2–3 requests/sec $\rightarrow$ 1 GPU (\$1–2/hour on-demand)

Total: $\sim$\$50/month base + compute as used.

Comparison: GPT-4 with 1M context $\sim$\$60/million input tokens. GMM becomes cost-competitive at $>$1000 requests/month for long-context tasks.

C Notation Summary

Symbol	Meaning
N	Total number of engrams
d	Embedding dimension
$\mathbb{S}^{d-1}$	Unit hypersphere in $\mathbb{R}^d$
$\phi(k) = (\mathbf{u}_k, r_k)$	Coordinate function
$\alpha = (\sqrt{p_1}, \ldots)$	Irrational basis (square roots of primes)
γ	Polynomial decay exponent
β_1, β_2	Layer compression factors
k_{fovea}	Foveal boundary
$\mathrm{H}(A_t)$	Attention entropy
λ	Signal loss weight

Table 2: Notation used throughout this paper.