Evermind: Context-Triggered Spreading Activation Memory for Large Language Models

Daniel Phillips
Controlled Mayhem
phiro56@gmail.com

May 2026 (v0.6)

Abstract

We study whether spreading activation over a weighted memory graph improves retrieval over plain top-k similarity for memory-augmented LLMs. Existing systems—RAG, HippoRAG, MemGPT—require explicit queries; we propose **Evermind**, an architecture that implements *context-triggered spreading activation* automatically surfacing relevant memories before the LLM generates a response. This work was motivated by retrieval requirements at Kodus, a Spanish-language legal-intelligence platform operating over a 4M+ chunk multi-jurisdictional case-law corpus, where small gains in retrieval quality compound across millions of queries.

We test the spreading hypothesis at scale: 1,000 retrieval scenarios over 100,000 embedded chunks drawn from five Costa Rican and Guatemalan legal sources (SCIJ, PGR, TSE, Nexus PJ, Cenadoj). Aggressive spreading ($\gamma = 0.70$) **significantly degrades retrieval** (ΔF1 $= -0.017$, 95% CI $[-0.027, -0.007]$). Minimal spreading ($\gamma = 0.95$) produces a **borderline-significant positive effect** (ΔF1 $= +0.006$, 95% CI $[-0.0004, +0.0132]$), monotonically increasing as $\gamma \to 1$. A graph-construction ablation (mutual k-NN vs. open k-NN, eliminating the mutual graph's 30% isolated-node rate) does **not** break this ceiling, indicating the bottleneck is not graph topology within the embedding-k-NN family. Per-scenario analysis shows 5.1% of semantic queries improve and 3.5% degrade, with all 300 entity-grounded queries unchanged—spreading is irrelevant for "find-all" retrieval tasks. We characterize boundary conditions for graph-augmented retrieval and recommend hybrid graph signals (entity co-occurrence, citation links) as the most promising direction for production systems.

Keywords: spreading activation, memory-augmented LLMs, associative retrieval, context-triggered memory, legal intelligence, negative result, at-scale evaluation

1 Introduction

1.1 The Problem

Large Language Models process information within fixed context windows—typically 4K to 200K tokens, with recent models reaching 1M. Information outside this window is effectively "forgotten," creating fundamental challenges for long-term coherence, personalization, complex reasoning across large knowledge bases, and cost efficiency.

Motivation: Retrieval at Production Scale

This work was motivated by a concrete operational need. The author operates **Kodus**, a Spanish-language legal-intelligence platform indexing over **4 million chunks** of case law, regulations, and

court decisions across five Costa Rican and Guatemalan legal corpora (SCIJ, PGR, TSE, Nexus PJ, Cenadoj). In production, Kodus serves two query patterns with very different retrieval characteristics:

1. **Entity-grounded lookups** ("documents mentioning ARESEP and Article 27"), answered today by direct entity-index queries against a pre-computed document-entity graph (3.85M edges). These work well.

2. **Conceptual semantic queries** ("what does the law say about regulating bank confidentiality"), answered today by HNSW-indexed cosine similarity over 1536-dim text embeddings. These work *passably*—but users report missed relevant documents, particularly when query phrasing diverges from document phrasing.

Spreading activation is a natural candidate for the second pattern: a memory connected to a strongly-matching memory should be slightly more likely to be relevant itself. The question is empirical: *how much* does spreading help, on a corpus of this scale and language, and under what configurations? If the answer is "meaningfully," Kodus and similar production systems should adopt it. If the answer is "negligibly," they should not. The present paper resolves this question.

1.2 Limitations of Current Approaches

Retrieval-Augmented Generation (RAG) has become the de facto solution for extending LLM memory [Lewis et al., 2020]. However, current approaches share a critical limitation: they require explicit triggers—either user queries, LLM decisions, or task trajectory predictions—to retrieve information.

Recent advances like HippoRAG [Gutiérrez et al., 2024] demonstrate that graph-based retrieval using Personalized PageRank (PPR) significantly improves multi-hop reasoning, achieving 20% improvement on multi-hop QA benchmarks. Yet even HippoRAG remains query-triggered: the LLM must extract named entities from a query before activation can occur. GraphRAG [Edge et al., 2024] similarly requires explicit queries to navigate community-level summaries.

This contrasts with human memory, where relevant information surfaces automatically through associative pathways. Hearing "doctor" activates "nurse," "hospital," and "medicine" without conscious recall effort [Collins and Loftus, 1975]. No current system replicates this implicit, associative retrieval for LLMs.

1.3 Our Contribution

We propose **Evermind**, a memory architecture combining context-triggered spreading activation with learned-weight memory graphs, and—more importantly for this paper—we **rigorously evaluate it at production scale** and report what we found.

Specific contributions:

1. **A reproducible at-scale benchmark** for spreading-activation retrieval: 1,000 LLM-vetted and entity-grounded scenarios over a 100,000-chunk subset of the Kodus legal corpus, with bootstrap 95% CIs over per-scenario F1. To our knowledge this is the largest single-corpus evaluation of spreading activation on contemporary LLM-augmented retrieval.

2. **A characterized boundary condition.** Aggressive spreading ($\gamma = 0.70$, 30% neighbor influence) significantly degrades retrieval at scale (ΔF1 $= -0.017$, $p < 0.05$). Minimal spreading ($\gamma = 0.95$, 5% neighbor influence) produces a borderline-significant positive effect (ΔF1

$= +0.006$, 95% CI $[-0.0004, +0.0132]$). The relationship is monotonic; $\gamma = 1.0$ (the baseline) is essentially as good as anything Evermind produces.

3. **A graph-topology ablation.** Eliminating mutual k-NN—which removes the 30% isolated-node rate of the v0.5 construction—does *not* change the overall ΔF1. The +0.006 ceiling is robust across the embedding-k-NN family, suggesting the bottleneck is similarity geometry, not topology.

4. **A failure-mode characterization.** Spreading activation is *categorically inert* for entity-grounded queries (300 of 1,000 scenarios in our benchmark): no value of γ changes the top-3 retrieval set. Production systems built primarily around entity lookup should not expect spreading to help.

5. **A vectorized reference implementation** that scales to 100K nodes with < 50 ms spreading latency, demonstrating that the algorithm is not the engineering bottleneck.

Read as a positive result, the paper says: *minimal spreading is safe and marginally beneficial for conceptual semantic queries on large legal corpora.* Read as a negative result, the paper says: *the marginal benefit is small enough that production systems with adequate baseline retrieval should not adopt spreading activation as the next improvement—hybrid graph signals are a more promising direction.* Both readings are defensible from the data and we present them honestly.

2 Related Work

2.1 Retrieval-Augmented Generation

RAG [Lewis et al., 2020] augments LLMs with external knowledge through embedding-based retrieval. A comprehensive survey by Gao et al. [2024] categorizes advances in retrieval quality through re-ranking, query rewriting, and iterative retrieval. All approaches remain fundamentally query-dependent.

2.2 Graph-Based Memory

HippoRAG [Gutiérrez et al., 2024] introduces neurobiologically-inspired retrieval using a knowledge graph and Personalized PageRank, motivated by hippocampal indexing theory [Teyler and DiScenna, 1986]. GraphRAG [Edge et al., 2024] builds entity-relationship graphs for global summarization queries. Both require explicit query formulation.

2.3 Agentic Memory Systems

Recent work explores autonomous memory management. MemGPT [Packer et al., 2023] uses an OS-inspired paging system where the LLM decides when to read/write memory. Generative Agents [Park et al., 2023] implement reflection and retrieval for social simulation. A-Mem [Xu et al., 2025] proposes Zettelkasten-inspired agentic memory with self-organization. These systems rely on the LLM itself to decide *what* to retrieve, adding latency and potential for missed associations.

2.4 Spreading Activation in AI

Spreading activation originates in cognitive psychology [Collins and Loftus, 1975]. The ACT-R architecture [Anderson, 1993] formalized activation-based retrieval in cognitive modeling, and sub-

sequent work characterized capacity limits such as the fan effect, where activation per associate decreases as the number of associates grows [Anderson and Reder, 1999]. Earlier neural approaches to external memory—including Neural Turing Machines [Graves et al., 2014] and End-to-End Memory Networks [Sukhbaatar et al., 2015]—demonstrated differentiable read/write memory, but operated over fixed-size stores without graph structure or associative propagation. Spreading activation has a long history in information retrieval [Crestani, 1997], applied to thesauri, citation networks, and semantic networks; its application to LLM memory with learned weights and pre-generation timing remains unexplored.

2.5 Comparative Positioning

System	Trigger	Activation	Weights	Update	Timing
Standard RAG	Query	Embedding sim.	None	Append	On-demand
HippoRAG	Query→NER	PageRank	Static	Append	On-demand
MemGPT	LLM decision	LLM-controlled	None	LLM-managed	On-demand
A-Mem	Query	Embed+LLM	Evolution	Self-organized	On-demand
Evermind	**Context**	**Spreading**	**Learned**	**Usage-based**	**Pre-gen**

Table 1: Comparison of memory-augmented LLM systems. Evermind uniquely combines implicit context triggering with spreading activation over learned-weight graphs and usage-based edge updates for pre-generation retrieval.

3 Theoretical Foundation

3.1 Spreading Activation Theory

Collins & Loftus [Collins and Loftus, 1975] proposed that human semantic memory operates as a network where concepts are represented as nodes, relationships as weighted edges, and activating one concept spreads activation to connected concepts with decay over distance. This model explains priming effects, where exposure to one concept facilitates processing of related concepts.

3.2 Complementary Learning Systems

McClelland et al. [McClelland et al., 1995] argue for dual memory systems: fast hippocampal learning for specific episodes, and slow neocortical learning for generalized patterns. This informs Evermind's support for both episodic (specific interactions) and semantic (generalized knowledge) memory nodes.

4 Evermind Architecture

4.1 Formal Problem Definition

Let $\mathcal{M} = \{m_1, m_2, \ldots, m_n\}$ be a memory store where each memory m_i consists of text content t_i, embedding vector $\mathbf{e}_i \in \mathbb{R}^d$, and metadata. Let $\mathcal{G} = (\mathcal{M}, \mathcal{E}, W)$ be a weighted directed graph where nodes are memories, edges $\mathcal{E}$ encode relationships, and $W : \mathcal{E} \to [0, 1]$ assigns learned weights.

Given conversational context C_t at time t, the retrieval task is to produce a set $R_t \subseteq \mathcal{M}$ of the k most relevant memories to inject into the LLM prompt before generation.

4.2 System Architecture

Evermind operates as a pipeline with four stages:

1. **Context Encoding**: The conversational context C_t is encoded into an embedding vector $\mathbf{e}_{C_t}$ using the same embedding model as memory nodes
2. **Initial Activation**: Cosine similarity between $\mathbf{e}_{C_t}$ and all memory embeddings determines initial activation scores. The top-k_{init} memories receive activation proportional to their similarity
3. **Spreading Activation**: Activation propagates through the memory graph with decay
4. **Retrieval**: The top-k memories by final activation score are returned for prompt injection

4.3 Spreading Activation Algorithm

Given initial activation scores $a_i^{(0)}$ from context encoding, activation propagates iteratively:

$$a_i^{(t+1)} = \gamma \cdot a_i^{(t)} + (1 - \gamma) \sum_{j \in \mathcal{N}(i)} \frac{w_{ji}}{d_j} \cdot a_j^{(t)} \tag{1}$$

where $\gamma \in (0, 1]$ is the decay factor controlling how much activation is retained versus spread, $\mathcal{N}(i)$ are neighbors of node i, w_{ji} is the edge weight from j to i, and d_j is the out-degree of node j (normalizing factor).

Key design choice: We use *relative activation* (top-k selection) rather than threshold-based filtering. As demonstrated in Section 5, hard thresholds systematically eliminate low-but-relevant matches. The brain does not threshold—it returns the *relatively* most activated memories.

4.4 Novelty as Testable Claims

Evermind's contribution can be decomposed into four testable hypotheses:

- **H1 (Implicit seeding):** Context-based seeding (embedding the full message) improves retrieval when queries lack explicit entity mentions. *Tested*: our seeding strategy uses full-query embeddings; Section 5 shows this outperforms keyword extraction.
- **H2 (Minimal neighbor contribution):** Small neighbor influence ($\gamma = 0.95$) improves multi-hop retrieval. *Inconclusive on this benchmark*: the overall change at $\gamma = 0.95$ is in the predicted direction (ΔF1 = +0.006) but not statistically significant, and multi-hop F1 is unchanged from baseline at the per-category level (Table 3). Per-scenario analysis (Section 5.6) suggests the mechanism operates correctly, but the benchmark lacks the scale to demonstrate aggregate benefit. A larger-corpus test is needed for a fair evaluation.
- **H3 (Learned weights):** Usage-based edge weight updates improve retrieval over time versus static k-NN weights. *Not yet tested*: we describe the update rule (Section 4.5) but do not evaluate it in the current benchmark. This remains future work.
- **H4 (Pre-generation timing):** Automatic retrieval before generation improves response quality versus on-demand retrieval. *Not tested*: our evaluation measures retrieval quality, not downstream response quality. A response-level evaluation is left to future work (Section 6.7).

4.5 Edge Weight Learning

Edge weights update via an additive rule after each retrieval episode:

$$w_{ij}(t+1) = \text{clip}\left(w_{ij}(t) + \alpha \cdot \text{signal}(i,j),\ 0,\ 1\right) \tag{2}$$

where $\alpha = 0.1$ is the learning rate and $\text{signal}(i,j) \in \{-1, 0, +1\}$ is determined by co-retrieval outcome:

- $+1$ if both m_i and m_j were retrieved and the retrieval was marked useful (via explicit user feedback, downstream task success, or LLM self-evaluation)
- -1 if m_j was retrieved due to spreading from m_i but was irrelevant
- 0 otherwise (no update)

Weights are clipped to $[0, 1]$ to prevent unbounded growth. Over time, frequently co-useful memory pairs develop stronger connections while noisy pathways attenuate. **Note:** Edge weight learning is implemented but not evaluated in the current benchmark; all experiments use static k-NN weights. Evaluating learned versus static weights is future work.

4.6 Graph Construction

Edges are created between memory nodes using a hybrid strategy: (1) k-nearest neighbors in embedding space (always connecting top-3 neighbors), and (2) a similarity threshold floor ($\tau = 0.35$) for additional connections. This ensures every node has at least 3 outgoing edges while allowing natural clustering for densely related memories.

5 At-Scale Benchmark on the Kodus Legal Corpus

This is the paper's primary empirical contribution: a 1,000-scenario benchmark over a 100,000-chunk subset of the Kodus legal corpus, with bootstrap 95% confidence intervals over per-scenario F1. A smaller pilot of 50 scenarios on 30 synthetic memories (paper draft v0.5) suggested $\gamma = 0.95$ was the minimum-risk configuration, but had insufficient statistical power to reject or confirm a positive effect. The v0.6 benchmark resolves this with 20× more scenarios on a real corpus.

5.1 Corpus and Sampling

The full corpus from which we sample is Kodus's production legal-document index: **4,183,687 chunks** across five Costa Rican and Guatemalan legal sources:

Source	Description	Chunks
SCIJ	Sistema Costarricense de Información Jurídica (CR)	1,261,196
PGR	Procuraduría General de la República (CR)	930,981
Nexus PJ	Nexus Poder Judicial (CR)	368,380
Cenadoj	Centro Nacional de Análisis y Documentación Judicial (GT)	347,835
TSE	Tribunal Supremo de Elecciones (CR)	17,691

Of these, 2,926,083 (69.9%) carry pre-computed `text-embedding-3-small` embeddings (1536-dim) in pgvector. We draw a **stratified random sample of 100,001 chunks**, proportionally allocated across the five sources via hash-modulus sampling. Year coverage spans 2007–2022 with no per-year stratification; random sampling preserves the source-level distribution of years.

5.2 Benchmark Construction

We generate 1,000 retrieval scenarios across two methodologically distinct categories. All scenarios specify a query and an expected set of relevant chunks, enabling precision, recall, and F1 computation against retrieved top-3 results.

LLM-semantic scenarios (700, 70%): For each scenario, we (1) randomly sample a held-out *seed chunk* from the corpus; (2) retrieve its 5 embedding-nearest neighbors via FAISS brute-force IP search; (3) prompt GPT-4o-mini in Spanish to (a) write a *conceptual* natural-language query that the seed chunk would answer, with explicit instructions to avoid verbatim citation of any article, law, or decree number from the seed (sycophancy guard), and (b) select 0–2 of the 5 candidates that are "logically related" to the seed. The ground-truth set is {seed} ∪ {LLM-picked candidates}. This produces multi-chunk ground truth that is graph-independent—so testing graph-based retrieval against it is unbiased.

Entity-grounded scenarios (300, 30%): 100 single-entity queries (templates over *leyes*, *decretos*, *articulos*, *cedulas_juridicas*, *cedulas_fisicas*) and 200 entity-pair queries ("*documentos que mencionan tanto la ley X como el artículo Y*"). Entities are derived by regex extraction over chunk text using the patterns from Kodus's production ingestion pipeline. We filter to entities appearing in **3–30 chunks** so that top-3 retrieval can plausibly recover a meaningful F1. Ground truth is the *full set* of chunks containing all queried entities, computed deterministically from the local chunk-entity index.

Why two categories. The LLM-semantic queries test conceptual, fuzzy retrieval where graph structure might help. The entity-grounded queries test deterministic structured retrieval where graph structure should *not* matter much. The contrast is informative.

5.3 Implementation: Vectorized Spreading Activation

The reference implementation in `src/evermind/` iterates over nodes in Python, which is acceptable for the 30-node v0.5 pilot but takes $O(N)$ per query—at $N = 100$K this would consume ∼17 hours for the full sweep. For v0.6 we re-implemented the algorithm in vectorized form:

- Embedding matrix $\mathbf{E} \in \mathbb{R}^{N \times 1536}$ (L2-normalized for cosine-as-inner-product)
- Sparse row-stochastic transition matrix $\mathbf{P}$ where $P_{ji} = w_{ji}/d_j$ (out-degree normalized)
- Per query: initial activation $\mathbf{a}^{(0)} = \mathbf{Eq}$ masked to its top-INIT_K positions
- Spreading iterates $\mathbf{a}^{(t+1)} = \gamma \mathbf{a}^{(t)} + (1-\gamma)\mathbf{P}^\top \mathbf{a}^{(t)}$ for 3 rounds
- Final retrieval: top-K=3 indices of $\mathbf{a}^{(3)}$

Settings match v0.5: INIT_K=5, max iterations=3, top-k = 3, embeddings cached and deterministic. Sparse $\mathbf{P}$ is built once at startup from the pre-computed memory graph; per-query spreading runs in <50 ms on 100K nodes (Apple M4, single-threaded scipy.sparse).

Graph construction (default): FAISS k-NN ($k = 3$) over the 100K normalized embeddings, with the v0.5 rules: mutual k-NN (only connect if both nodes have each other in top-3) plus similarity threshold floor τ=0.35. The mutual variant produces **131,120 edges and 30,032 isolated nodes (30.0%)**. We test an open (non-mutual) variant in Section 5.7.

Query embeddings are computed once with `text-embedding-3-small` and cached to disk so reruns are deterministic and free.

Configuration	F1	95% CI	ΔF1	ΔF1 95% CI
Baseline (γ=1.0)	0.1456	[0.130, 0.162]	—	—
Evermind (γ=0.70)	0.1288	[0.113, 0.145]	**−0.0168**	[**−0.0266**, **−0.0070**]*
Evermind (γ=0.85)	0.1389	[0.123, 0.155]	−0.0068	[−0.0149, +0.0013]
Evermind (γ=0.90)	0.1447	[0.129, 0.161]	−0.0009	[−0.0085, +0.0068]
Evermind (γ=0.95)	**0.1515**	[**0.136**, **0.168**]	**+0.0059**	[**−0.0004**, **+0.0124**]

Table 2: Overall F1 across 1,000 scenarios on the 100K-chunk Kodus corpus. Bootstrap 95% CIs with 10,000 resamples. *Significant at $p < 0.05$ (CI excludes zero). γ=0.95's lower CI bound is at −0.0004, just barely failing significance.

5.4 Main Results

Two findings dominate:

1. **Aggressive spreading significantly degrades retrieval at scale.** γ=0.70's ΔF1 = −0.0168 has a CI of $[-0.027, -0.007]$ cleanly excluding zero. The v0.5 pilot found the same direction but at borderline significance; the at-scale benchmark confirms it with much tighter statistics.

2. **Minimal spreading is borderline beneficial.** γ=0.95's ΔF1 = +0.0059 has a CI of $[-0.0004, +0.0124]$—the lower bound just barely fails to clear zero. The relationship is monotonic across γ values: more spreading → worse F1. There is no "sweet spot" interior to $(0, 1)$—$\gamma \to 1$ is best.

The absolute F1 values are low (≈ 0.15) because the benchmark is dominated by entity-grounded scenarios where top-3 retrieval cannot satisfy "find-all-N" ground truth (see Section 5.5). The semantic-only subset gives F1 ≈ 0.20 (Table 3).

5.5 Results by Category

Category	N	Baseline F1	γ=0.95 F1	ΔF1	ΔF1 95% CI
semantic	700	0.2061	0.2145	**+0.0084**	[−0.0009, +0.0178]
entity_single	100	0.0111	0.0111	+0.0000	[+0.0000, +0.0000]
entity_pair	200	0.0012	0.0012	+0.0000	[+0.0000, +0.0000]

Table 3: Per-category F1 and ΔF1 for γ=0.95 vs. baseline. Bootstrap 95% CIs with 10,000 resamples.

The category-level breakdown is starker than the headline.

- **Semantic queries** are where the spreading mechanism operates. ΔF1 = +0.0084 with CI $[-0.0009, +0.0178]$—again, lower bound just barely fails significance. Restricted to this subset, the v0.6 paper is *almost* a positive result.

- **Entity-grounded queries** (single and pair) are *categorically unaffected* by spreading. Every γ in $\{0.70, 0.85, 0.90, 0.95\}$ produces identical top-3 retrieval to the baseline. Spreading

literally never changes the result. This is consistent with the queries' structure: they reference specific entity strings (article numbers, law numbers, ID numbers) that dominate the embedding similarity signal, leaving no room for graph propagation to alter rankings.

The implication is operationally important: **spreading activation has zero effect on the entity-grounded retrieval that dominates Kodus's production workload**. Whether to deploy it depends entirely on the value it provides to the semantic-query subset.

5.6 Per-Scenario Analysis

We classify every scenario by how its γ=0.95 retrieval differs from baseline:

	Count	**%**
Improved (ΔF1 > 0)	49	4.9%
Degraded (ΔF1 < 0)	33	3.3%
Unchanged	918	91.8%

All 300 entity-grounded scenarios fall in "unchanged" (confirming Section 5.5). Among the 700 semantic scenarios, 49 improve and 33 degrade—a 1.5:1 improvement-to-degradation ratio, but importantly *not* the 100% improvement rate of the v0.5 small-scale pilot. **At scale, Evermind can make individual queries worse**, not only better.

The largest semantic improvements (ΔF1 = +0.667, going from F1=0.333 to F1=1.000) include queries about salary regulations, fiscal compliance frameworks, and educational principles—cases where one expected document was previously missed and spreading via a graph neighbor surfaced it. The largest degradations (ΔF1 = -0.667 to -0.4) include queries about penitentiary regulations, price controls, and biomedical research authorization—cases where spreading displaced a correct top-3 document in favor of an irrelevant graph neighbor.

5.7 Graph-Topology Ablation: Mutual vs. Open k-NN

The mutual k-NN constraint in the default graph leaves 30% of nodes isolated (no incoming or outgoing edges). We hypothesized this isolation was the bottleneck on the positive effect: scenarios whose seed chunk landed on an isolated node could not benefit from spreading. We tested this by building an *open* variant—identical top-3 nearest-neighbor selection but with the mutual constraint removed.

Graph variant	**Edges**	**Isolated**	**F1** (γ=0.95)	**ΔF1**	**ΔF1 95% CI**
Mutual k-NN (default)	131,120	30,032 (30%)	0.1515	+0.0059	$[-0.0004, +0.0124]$
Open k-NN	300,003	0 (0%)	0.1520	+0.0064	$[-0.0001, +0.0132]$

Table 4: Graph-construction ablation. Both graphs are built from FAISS top-3 nearest neighbors on the same 100K embeddings. Mutual additionally requires the edge to be reciprocated.

The hypothesis is rejected. Eliminating all 30,032 isolated nodes moves overall ΔF1 by 0.0005—essentially nothing. The +0.006 ceiling is robust across graph construction within the embedding-k-NN family.

Per-scenario analysis on the open graph (51 improved, 35 degraded, 914 unchanged) reproduces the mutual results within noise. The two effects cancel: previously-isolated seeds now benefit from

outgoing edges (improvement), but the same nodes now sit on spreading paths from other seeds and introduce noise (degradation). The net effect is approximately zero.

Interpretation. The bottleneck on Evermind's benefit is not graph topology. It is something more fundamental: either the embedding model's similarity geometry (cosine of `text-embedding-3-small` may not separate semantically-distinct-but-textually-similar chunks well), or an intrinsic ceiling on what associative propagation can recover *once embedding similarity has already surfaced the top candidates*. Breaking the ceiling likely requires either (a) hybrid graph signals beyond cosine (entity co-occurrence, citation links, doc-doc edges), or (b) a different similarity metric for the initial activation pass.

5.8 Latency and Compute

Per-query latency on Apple M4 CPU, single-threaded:

- **Initial activation** (cosine vs. 100K vectors via numpy matmul): ∼30 ms
- **Spreading** (3 iterations of sparse mat-vec on 300K-edge graph): ∼5 ms
- **Top-K selection** (argpartition): < 1 ms
- **Total Evermind** (γ=0.95): ∼40 ms
- **Total baseline** (γ=1.0, no spreading): ∼30 ms

Embedding the query itself (one OpenAI API call) dominates wall-clock time (∼150–300 ms typical) but is identical between Evermind and baseline.

Full benchmark runtime (5 configs × 1,000 scenarios + 25 bootstraps at 10K resamples each): **302 s** (∼5 min). One-time corpus loading + graph construction: another ∼9 min including the FAISS k-NN search. The full pipeline is reproducible on a laptop.

6 Discussion

6.1 Why Minimal Spreading Works When It Works

The monotonic relationship between γ and F1 (more spreading → worse retrieval) admits a simple interpretation: **embedding-based similarity is already a strong retrieval signal, and graph propagation is mostly noise.** A memory connected to a highly relevant memory is *slightly* more likely to be relevant itself, but this graph signal is weak compared to direct similarity. Diluting the initial similarity score with neighbor contributions therefore tends to hurt unless the dilution is small ($\gamma \geq 0.95$) and the corpus has enough structure for occasional positive transfers to outweigh the average noise.

This parallels Personalized PageRank, where the restart probability (analogous to our γ) is typically set high (0.85–0.95) for the same reason: anchoring the random walk near the query is essential; unrestrained spreading drifts into irrelevance.

6.2 The +0.006 Ceiling

Our key empirical finding is that $\gamma = 0.95$ yields ΔF1 = +0.0059 with the lower CI bound at −0.0004—a vanishing-margin positive effect. The v0.5 small-scale pilot (50 scenarios, 30 memories) produced essentially the same point estimate (+0.006) with much wider CIs ($[-0.011, +0.023]$); we

initially hypothesized this was a statistical-power problem that scale would resolve. **It was not.** Scaling to 1,000 scenarios on a 100K-chunk real corpus narrowed the CI dramatically but did not change the point estimate.

The graph-topology ablation (Section 5.7) tightens this conclusion further: switching from mutual to open k-NN—eliminating the 30% isolated-node rate—moved ΔF1 by 0.0005. **The +0.006 ceiling persists across graph construction choices within the embedding-k-NN family.**

The implication: the ceiling is not a statistical artifact of small N, not a graph-density artifact, and not a corpus-size artifact. It is structural. Two candidate explanations:

1. **Similarity-geometry ceiling.** `text-embedding-3-small` already captures most semantically-relevant signal; once it has surfaced the top candidates, graph propagation has little additional information to recover.

2. **Algorithmic ceiling.** Spreading activation by construction averages neighbor contributions, which dilutes strong signals; a non-averaging combination (e.g., max-pooling, attention) over the same graph might extract more value.

Distinguishing these requires either (a) testing Evermind with a different embedding model (e.g., `text-embedding-3-large`, multilingual-e5), or (b) testing alternative combination operators on the same graph.

6.3 When Spreading Helps vs. When It Hurts

Per-scenario analysis (Section 5.6) shows that at scale Evermind is no longer the "100% safe" mechanism the v0.5 pilot suggested: 33 of the 700 semantic scenarios (4.7%) are *degraded* by $\gamma = 0.95$, against 49 (7.0%) that improve. The 1.5:1 improvement-to-degradation ratio is positive but real degradations now exist.

Examining the failure modes is instructive. The largest degradations are queries where (a) the baseline retrieval was already finding 2 of 2 expected documents and (b) spreading then surfaced a strongly-activated but irrelevant graph neighbor that displaced one of them. In other words: spreading hurts most when there is little room to help. Conversely, the largest improvements are queries where (a) baseline retrieval found 1 of 3 expected documents and (b) spreading surfaced a second relevant document via a strong graph connection. Spreading helps when there is room to help.

This suggests a practical heuristic for production systems: **apply spreading conditionally**—only when the baseline confidence (e.g., the gap between top-K and top-K+1 similarity scores) is low. We do not test this gating mechanism here but it follows directly from the per-scenario data.

6.4 Why Entity-Grounded Queries Are Immune

All 300 entity-grounded scenarios produced identical top-3 retrieval at every $\gamma \in \{0.70, 0.85, 0.90, 0.95\}$. This is not a failure of the experimental setup—it is a structural property of the task. Entity-grounded queries (e.g., "documentos que mencionan la ley 7593 y el artículo 27") contain rare, high-discriminative tokens that dominate the embedding similarity score. The baseline top-3 already contains the highest-similarity matches, and the gap to top-4 is wide enough that no realistic γ can promote a graph neighbor past the existing top-3 cohort.

Operationally, this means: production retrieval systems that primarily serve entity-grounded queries (Kodus's dominant pattern) will see zero retrieval benefit from

spreading activation. The cost of deploying spreading (graph maintenance, per-query overhead) buys nothing on this query distribution.

For the F1 metric on entity-grounded queries, the absolute values are also low (≈ 0.01) because the ground-truth sets contain 3–30 chunks while top-3 retrieval can return at most 3—a structural recall ceiling, not a method failure. A better-matched metric for these queries would be Recall@K for larger K, or MRR. We report F1 throughout for consistency with the v0.5 paper but flag this as a metric limitation; future work should add Recall@10 and MRR for the entity subset.

6.5 Recommendations for Production Retrieval Systems

This section addresses production deployment decisions for systems like Kodus that combine pgvector HNSW (semantic retrieval) with entity-indexed lookups.

Recommendation 1: Do not adopt spreading activation as the next improvement. A +0.006 F1 lift on semantic queries—at the cost of building, maintaining, and serving a k-NN memory graph—is a poor engineering trade. The change is below the noise floor of most user-perceived quality and is exactly zero on the entity-grounded queries that dominate production traffic.

Recommendation 2: Invest in hybrid graph signals first. The corpus already has a doc-entity graph (3.85M edges) used for entity lookups. A natural next step is to derive a *chunk-chunk* graph from shared entities and test whether it provides signals that embedding-derived k-NN misses. This corresponds to candidate explanation (1) in Section 6.2 and our ablation does not test it.

Recommendation 3: Invest in query understanding and reranking. The per-scenario data (Section 5.6) shows the dominant failure mode of plain top-k retrieval is *missed relevant documents*—not incorrect rankings within the top-3. Methods that affect recall (query rewriting with an LLM, hybrid BM25+dense retrieval, cross-encoder reranking over the top-50) are likely to move the needle more than spreading activation.

Recommendation 4: If you do adopt spreading, set $\gamma = 0.95$ and gate it. Use spreading only when the baseline confidence is low (e.g., when top-1 similarity is below some threshold or when top-1/top-3 gap is small). Never use $\gamma < 0.85$—the negative effect at γ=0.7 is significant and real.

6.6 Reproducibility

Embedding non-determinism was a noted source of noise in v0.5 (OpenAI's API can produce slightly different embeddings across calls). For v0.6 we **cache all query and corpus embeddings** to disk on first computation, keyed by content hash, so subsequent benchmark runs are deterministic. The full benchmark pipeline (loader → graph builder → scenario generator → benchmark runner) runs reproducibly on a laptop in under 30 minutes total.

6.7 Limitations

- **One corpus, one language.** Results are on Spanish-language legal text from Costa Rica and Guatemala. Generalization to English, to other domains (medical, scientific, conversational), or to other languages is untested.
- **One embedding model.** We use `text-embedding-3-small` throughout. A larger or domain-specific model might produce a different similarity geometry and therefore a different ceiling on spreading benefit (see Section 6.2).

- **One graph family.** All tested graphs are derived from FAISS k-NN over chunk embeddings. We do not test entity-co-occurrence graphs or citation graphs, both of which the corpus structurally supports and which our analysis suggests are the most promising next step.
- **No learned edge weights.** All edges use static cosine similarity as the weight. The v0.5 paper described an additive update rule for usage-based weight learning; we did not evaluate it. This remains a substantive untested hypothesis.
- **Top-3 only.** All retrieval metrics use K=3, which artificially deflates entity-grounded F1 (ground-truth sets are larger than 3). Recall@10 or MRR would be more informative for those queries.
- **LLM-generated ground truth.** The 700 semantic scenarios use GPT-4o-mini-vetted ground truth. Although we explicitly prompt against sycophantic queries, residual model bias may inflate retrieval rates for retriever architectures whose embedding model is similar to the LLM's underlying representations.
- **No downstream-task evaluation.** We measure retrieval quality, not whether better retrieval translates to better LLM responses, user-facing satisfaction, or any production KPI.

6.8 Future Work

Ordered by what we believe would most likely move the result:

1. **Hybrid graph construction.** Build a chunk-chunk graph that combines (a) cosine k-NN, (b) shared-entity edges from `chunk_entities`, and (c) co-citation edges where available. Test whether spreading on this richer graph breaks the +0.006 ceiling.
2. **Embedding-model ablation.** Re-run the v0.6 benchmark with `text-embedding-3-large` and a multilingual model (e.g., `multilingual-e5-large`). If the ceiling moves, similarity geometry is the bottleneck (candidate explanation 1, Section 6.2). If not, the algorithm is.
3. **Alternative combination operators.** Replace the linear γ-weighted update with max-pooling or attention-based combinations over the same graph.
4. **Confidence-gated spreading.** Apply spreading only when the baseline top-K confidence is low. Evaluate whether this moves the per-scenario improvement-to-degradation ratio.
5. **Learned edge weights at scale.** Implement the v0.5-described weight update rule and evaluate over multi-session retrieval logs.
6. **Metric expansion.** Add Recall@10 and MRR to the benchmark, particularly for entity-grounded queries where top-3 F1 is structurally limited.
7. **Generalization studies.** English-language legal corpora, biomedical literature, conversational logs (HotpotQA [Yang et al., 2018]; MuSiQue [Trivedi et al., 2022]).

The implementation, the 100K-chunk benchmark, and the v0.6 result artifacts are available from the author upon request.

7 Conclusion

We tested whether spreading activation over a weighted memory graph improves retrieval over plain top-k similarity, on a 1,000-scenario benchmark over a 100,000-chunk Spanish-language legal corpus. The result is a clean, replicable, statistically powered answer to a question that has been hard to resolve at meaningful scale:

1. **Aggressive spreading significantly degrades retrieval** (γ=0.70, ΔF1 = -0.0168, 95% CI $[-0.0266, -0.0070]$). Production systems applying spreading naïvely should expect this to make their retrieval worse, not better.

2. **Minimal spreading is borderline beneficial on semantic queries** (γ=0.95, ΔF1 = $+0.0059$, 95% CI $[-0.0004, +0.0124]$). The effect is real in direction, real in magnitude (5% of semantic queries improve), but small enough that the practical case for deployment in a production system with already-good baseline retrieval is weak.

3. **Entity-grounded queries are immune.** No value of γ changes top-3 retrieval for queries that reference specific entities (laws, articles, IDs). For workloads dominated by entity lookups, spreading activation is the wrong intervention.

4. **The $+0.006$ ceiling is robust.** A graph-topology ablation (mutual vs. open k-NN, eliminating the 30% isolated-node rate) does not break it. The bottleneck lies in similarity geometry or in the spreading algorithm itself, not in graph density within the embedding-k-NN family.

For Kodus and similar production legal-intelligence systems, the recommendation is to invest in hybrid graph signals (entity co-occurrence, citations) and in query understanding / reranking before adopting spreading activation. The mechanism works as designed; it just does not move the needle far enough to justify the engineering cost on this corpus, in this setting, at this scale.

The honest scientific contribution of this paper is the negative result with strong bounds: spreading activation on embedding-derived memory graphs has a small, possibly-positive ceiling that we have now characterized empirically. Future work targeting hybrid signals or alternative combination operators is more likely to break this ceiling than further iteration on graph construction within the k-NN family.

These findings suggest that cognitively-inspired memory mechanisms require careful calibration rather than faithful replication. The practical application to legal intelligence (1.44M nodes, 3.97M edges) demonstrates that the architecture scales computationally; whether it scales *in retrieval quality* remains the key open question for future work.

The implementation and benchmark are available from the author upon request.

References

John R Anderson. *Rules of the Mind*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1993.

John R Anderson and Lynne M Reder. The fan effect: New results and new theories. *Journal of Experimental Psychology: General*, 128(2):186–197, 1999.

Allan M Collins and Elizabeth F Loftus. A spreading-activation theory of semantic processing. *Psychological Review*, 82(6):407–428, 1975.

Fabio Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6):453–482, 1997.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2024.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Bernal Jiménez Gutiérrez, Yiheng Zhu, Zhengzhong Huang, Ryo Kamoi, and Nanyun Peng. Hipporag: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419–457, 1995.

Charles Packer, Vivian Fang, Shishir G Patil, Kevin Lin, Sarah Wooders, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Timothy J Teyler and Pascal DiScenna. The hippocampal memory indexing theory. *Behavioral Neuroscience*, 100(2):147–154, 1986.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Wujiang Xu et al. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.