Vrin: From Retrieval to Reasoning — A Hybrid Knowledge Graph Architecture for Enterprise AI

Vedant Patel[a,1]

[a]Vrin, https://vrin.cloud

April 2026

Retrieval-Augmented Generation (RAG) has become the standard approach for grounding large language model responses in factual data. Yet current implementations remain fundamentally limited: they retrieve text chunks through semantic similarity and rely on the LLM to reason over unstructured fragments. This approach fails on queries requiring multi-document reasoning, temporal awareness, or numerical constraints. We present Vrin, a hybrid knowledge graph architecture that transforms RAG from a retrieval mechanism into a reasoning engine. Vrin introduces a multi-stage pipeline comprising entity-centric fact extraction with coreference resolution, temporal fact versioning with bi-temporal conflict resolution, knowledge consolidation with community detection and cross-fact deduplication, knowledge-aware query planning that aligns retrieval strategy with the graph's topology, confidence-scored multi-hop graph traversal with Personalized PageRank complementary retrieval and parallel reasoning strategies, iterative reasoning with per-step quality evaluation, adaptive query complexity routing, constraint-aware retrieval across five types, hybrid BM25 and vector search with reciprocal rank fusion, multi-dimensional retrieval confidence assessment with adaptive bail-out, and structured context preparation that organizes evidence by concept rather than by source. On MultiHop-RAG, Vrin achieves 95.1% accuracy, outperforming GPT 5.2 (78.9%) with the same evidence documents. On MuSiQue, a multi-hop QA benchmark designed to resist reasoning shortcuts, Vrin achieves 0.478 Exact Match and 0.563 Token F1, surpassing the state-of-the-art HippoRAG 2 (0.372 EM, 0.486 F1) on both metrics. These results demonstrate that the gap between retrieval and reasoning is not a limitation of LLMs, but of the architectures surrounding them.

Retrieval-Augmented Generation | Knowledge Graphs | Multi-Hop Reasoning | Enterprise AI | Hybrid Search | Cognitive Architecture

The Retrieval-Augmented Generation paradigm has achieved remarkable adoption since its introduction Lewis et al. (2020). By 2026, RAG underpins most enterprise AI deployments, from customer support chatbots to financial analysis tools. Yet a striking paradox persists: the vast majority of enterprise AI pilots fail to deliver measurable ROI. Organizations invest in vector databases, embedding pipelines, and prompt engineering, only to find that their AI systems cannot reliably answer questions that a junior analyst could handle with a spreadsheet and ten minutes of reading.

The root cause is architectural. Current RAG systems perform one operation exceedingly well (semantic similarity search) and rely on the LLM to do everything else. When a user asks *"How did TechCorp's revenue change after the CEO transition in Q3?"*, a standard RAG system finds the most semantically similar text chunks and feeds them to the model. It does not identify the entities involved, the temporal constraint, the causal relationship, or the comparison being requested. All of these reasoning steps are delegated to the LLM as an implicit, unstructured task.

This is not how reasoning works. Consider how a human analyst would approach the same question through five distinct cognitive subprocesses:

1. **Perceive**: Identify the key entities and relationships
2. **Structure**: Formalize the question as a constrained search
3. **Store**: Know where the relevant information lives
4. **Organize**: Connect related facts across documents
5. **Retrieve**: Pull the specific facts needed

Each subprocess is a distinct engineering challenge. The current RAG paradigm addresses only the last one, and does so imprecisely through semantic matching rather than structured retrieval. The first four subprocesses are entirely absent.

These subprocesses are not arbitrary divisions. They map to established constructs in cognitive science: the brain's dual-store architecture (hippocampus for fast episodic indexing, neocortex for slow structured knowledge) described by Complementary Learning Systems theory Kumaran et al. (2016), semantic network theory Collins and Loftus (1975), and metacognitive monitoring Fleming (2024). In 2024, HippoRAG Gutierrez et al. (2024) explicitly applied hippocampal memory theory to RAG at NeurIPS, building a graph-plus-vector hybrid that outperformed standard RAG by up to 20% on multi-hop questions. Vrin independently arrived at the same architecture—a convergence that suggests these engineering problems have a natural solution space.

Vrin is built on the thesis that *each of these five cognitive subprocesses must be engineered explicitly.* Rather than treating retrieval as a single undifferentiated step, Vrin implements a multi-stage reasoning pipeline where knowledge is perceived through entity-centric extraction, structured as typed facts in a knowledge graph, stored with temporal metadata and confidence scores, organized through conflict detection and version management, and retrieved through constraint-aware multi-hop traversal fused with hybrid search.

The result is a system that doesn't just find relevant text; it reasons over structured knowledge. Vrin achieves 95.1% on MultiHop-RAG versus 78.9% for GPT 5.2 with the same evidence, and 0.478 Exact Match on MuSiQue Trivedi et al. (2022) versus HippoRAG 2's Gutierrez et al. (2025) 0.372 (+28%), surpassing the current state of the art on compositional multi-hop QA on both Exact Match and Token F1.

We believe the industry has explored less than 5% of the innovation space in knowledge-augmented AI, and the remaining 95% lies not in better embeddings or larger context windows, but in the engineering of perception, structure, storage, and organization.

Significance Statement

Current RAG systems delegate reasoning to the LLM, failing on multi-hop, temporal, and numerical queries. Vrin's hybrid knowledge graph achieves 95.1% on MultiHop-RAG vs. GPT 5.2's 78.9%, and 0.478 EM on MuSiQue vs. HippoRAG 2's 0.372—a 28% improvement over the state of the art.

V.P. designed and implemented the Vrin system, conducted all experiments, and wrote the paper.

The author is the founder of Vrin. Benchmark evaluation code is open-source.

[1]To whom correspondence should be addressed. E-mail: vedant@vrin.cloud

1. Related Work

A. Traditional RAG.. The original RAG framework Lewis et al. (2020) established the retrieve-then-generate paradigm. Subsequent work improved individual components, including better embeddings Neelakantan et al. (2022), smarter chunking Anthropic (2024), and more effective reranking Nogueira and Cho (2019), but the fundamental architecture remains unchanged: the query is a bag of semantics, the knowledge base is a bag of chunks, and the LLM bridges the gap.

B. Knowledge Graph Approaches.. Microsoft's GraphRAG Edge et al. (2024) introduced community summaries built from knowledge graphs, enabling global queries over document collections. However, GraphRAG lacks temporal awareness, constraint handling, or real-time hybrid search. T-GRAG Li et al. (2025a) addressed temporal reasoning, concurrent with Vrin's temporal fact versioning. T-GRAG focuses specifically on temporal queries, while Vrin's temporal system is one component of a broader multi-stage pipeline.

C. Reasoning-Enhanced RAG.. Self-RAG Asai et al. (2024) teaches models to self-reflect on retrieval quality. CoT-RAG Li et al. (2025b) integrates chain-of-thought with retrieval decisions. DeepRAG Guan et al. (2025) models retrieval as a Markov Decision Process, making adaptive decisions about when to retrieve. These systems advance reasoning in RAG but typically focus on a single dimension rather than addressing the full perception-to-retrieval pipeline.

D. Entity-Centric Retrieval.. Entity Similarity RAG Wang et al. (2025) introduced entity-centric retrieval using entity similarity for knowledge graph construction. Vrin shares the intuition that entities should be first-class objects but extends it with coreference resolution, temporal versioning, constraint handling, and hybrid graph-vector fusion.

E. Cognitive Architecture and Neuroscience.. The five-subprocess framework in Section 1 maps to established constructs in cognitive science. CLS theory Kumaran et al. (2016) proposes that the brain uses a dual-store architecture: the hippocampus for fast episodic indexing and the neocortex for slow, structured knowledge consolidation. HippoRAG Gutierrez et al. (2024) explicitly applied this theory to RAG, achieving up to 20% improvement on multi-hop questions. HippoRAG 2 Gutierrez et al. (2025) extended this with a dual-node knowledge graph combining passage and phrase nodes, enhanced by Personalized PageRank and LLM-based triple filtering, establishing the current state of the art on MuSiQue multi-hop QA. Vrin independently arrived at the same dual-store architecture (Neptune graph plus OpenSearch vectors) and extends it with temporal versioning, confidence scoring, adaptive query complexity routing, and enterprise data sovereignty.

The brain's knowledge representation is itself a graph: semantic network theory Collins and Loftus (1975) describes entity-relationship structures in human memory, and recent work has confirmed this at the synaptic level, showing that memory engrams organize through hub-like multi-synaptic structures rather than point-to-point connections Uytiepo et al. (2025). The brain's metacognitive monitoring system, mediated by the anterior cingulate cortex Fleming (2024), detects retrieval uncertainty and can halt the process when evidence is insufficient—a direct analog to Vrin's adaptive bail-out (Section 3, Stage 4). Webb et al. Webb et al. (2025) demonstrated that brain-inspired modular architectures outperform monolithic LLMs on planning tasks, and Whittington et al. Whittington et al. (2022) formally showed that spreading activation in neural semantic networks is mathematically equivalent to transformer attention. Bakermans et al. Bakermans et al. (2025) showed that the brain consolidates episodes into reusable knowledge through compositional replay—a process that maps to how Vrin's fact extraction pipeline transforms documents into structured graph knowledge.

F. Positioning.. Vrin is not the first system to use knowledge graphs for RAG, nor the first to integrate reasoning with retrieval. Its contribution is the *integration* of these capabilities into a unified, production-grade pipeline: entity-centric extraction, temporal versioning, five-type constraint solving, confidence-scored multi-hop traversal, hybrid search with rank fusion, cross-encoder reranking, and focused chain-of-thought, all operating together on every query.

2. Architecture

Vrin's architecture maps to the five cognitive subprocesses identified in Section 1. We describe each in order: the insertion pipeline (perceive and structure), the knowledge graph (store and organize), and the query pipeline (retrieve and reason). Figure 1 provides a high-level overview.

A. Knowledge Insertion Pipeline.. When a document enters Vrin, it passes through a multi-stage extraction pipeline that transforms unstructured text into structured, entity-linked facts.

Two-Phase Entity-Centric Extraction. Standard RAG systems chunk documents and embed the chunks. Vrin takes a fundamentally different approach: it extracts structured facts as subject-predicate-object triples, where subjects are always concrete, named entities.

The extraction operates in two phases. In **Phase 1** (Entity Identification), an LLM identifies the main entities in each text chunk: organizations, people, products, projects, and locations. Each entity is typed and scored for importance. In **Phase 2** (Entity-Linked Fact Extraction), the LLM extracts subject-predicate-object triples using the identified entities as context anchors.

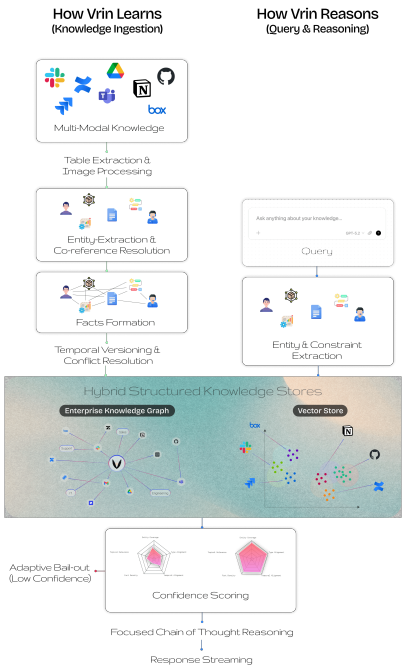

Fig. 1. Vrin system architecture. Documents enter through the insertion pipeline (left), where entity-centric extraction builds a knowledge graph with temporal versioning. Queries are processed through the multi-stage reasoning pipeline (right), combining graph traversal, PPR, hybrid search, confidence assessment, and focused chain-of-thought generation.

A critical requirement is **coreference resolution**: the system resolves pronouns and indirect references ("it," "the company," "they") to their concrete entity referents before creating facts. This ensures that "TechCorp announced earnings. It reported $245M in revenue" produces the triple `(TechCorp, reported_revenue, $245M)` rather than `(It, reported_revenue, $245M)`.

Attribute vertex prevention ensures literal values such as monetary amounts, percentages, and years are stored as edge properties rather than graph vertices. Without this, "$245M" and "2024" would become entity nodes, creating spurious connections across unrelated facts.

Every extracted fact carries a **confidence score** between 0.0 and 1.0. Facts below a configurable threshold are discarded. The extraction model, timestamp, and source document are recorded for full provenance.

Batch extraction with cross-chunk context. Rather than processing each chunk independently, Vrin batches consecutive chunks (up to 25 per LLM call) to provide cross-chunk context during extraction. This enables the model to resolve references that span chunk boundaries and extract relationships that no single chunk contains in isolation.

Content-hash deduplication. Each chunk is assigned an ID derived from its content hash, ensuring that re-ingesting the same content (e.g., periodic sync from a connected data source) updates the existing record rather than creating duplicates. This is critical for enterprise deployments where knowledge bases are continuously synchronized.

Table-Aware Processing. Financial and technical documents frequently contain tables. Standard chunking destroys table structure, making it impossible for the LLM to reason about row-column relationships. Vrin detects tables in multiple formats (markdown, HTML, JSON, CSV) and extracts them separately, preserving structural information needed for numerical reasoning.

Visual Intelligence. Vrin processes visual content using multimodal vision models. Charts are converted to structured JSON with axis labels and data points, transforming a bar chart into queryable facts. Diagrams are decomposed into nodes and edges. Cross-modal embeddings (1024 dimensions) place images and text in the same vector space, enabling text queries to discover relevant visual content.

B. Knowledge Graph with Temporal Versioning.. Vrin stores extracted knowledge in a property graph (Amazon Neptune) alongside vector embeddings in a search index (Amazon OpenSearch). This dual-store architecture enables both structured graph traversal and semantic similarity search, mirroring the brain's Complementary Learning Systems Kumaran et al. (2016) where the hippocampus provides fast episodic indexing and the neocortex maintains slow, structured representations—a parallel now directly applied to RAG by HippoRAG Gutierrez et al. (2024).

Graph Structure. The knowledge graph consists of typed entity vertices connected by fact edges. Each edge carries the predicate, confidence score, temporal validity window (`valid_from`, `valid_to`), status (active, superseded, corrected), numerical metadata (queryable numeric value, unit, and type), and source attribution. This structure enables queries that standard RAG cannot handle: "What was TechCorp's revenue in 2022?" becomes a graph traversal with a temporal filter.

Temporal Fact Versioning. Knowledge evolves. A company's CEO changes, revenue figures are updated quarterly. Standard

RAG treats all information as equally current, leading to contradictions. Vrin implements automatic temporal versioning:

- **Conflict detection**: New facts with the same subject-predicate but different objects trigger conflict resolution
- **Supersession**: Older facts are closed with `valid_to` set and status changed to "superseded"; both versions remain linked
- **Bi-temporal tracking**: Each fact carries both an event time (`valid_from`/`valid_to`) and an ingestion time (`ingested_at`), enabling the system to distinguish "when did this happen?" from "when did we learn this?"—critical for audit trails and late-arriving corrections
- **Adaptive granularity**: Temporal resolution adapts to update frequency: day-level for rapidly changing facts, year-level for infrequent changes
- **Time-travel queries**: Temporal metadata enables point-in-time retrieval

C. Knowledge Consolidation.. As a knowledge graph grows, redundancy and inconsistency accumulate. Vrin runs a periodic consolidation pipeline that maintains graph quality through four operations:

Community detection. The Leiden algorithm Traag et al. (2019) identifies clusters of densely connected entities, partitioning the knowledge graph into thematic communities (e.g., a cluster of agent memory research, a cluster of evaluation methodology). These community assignments are stored on graph vertices and used during query planning to identify which knowledge clusters are relevant to a given question.

Cross-fact deduplication. A three-stage cascade identifies duplicate facts: (1) structural blocking groups facts by normalized subject-predicate pairs, (2) fuzzy matching scores candidate pairs using sequence similarity, and (3) LLM verification confirms or rejects near-duplicates. Confirmed duplicates are merged using Noisy-OR confidence combination Dong et al. (2014), preserving provenance from both sources.

Graph-aware ingestion. When new documents are ingested, extracted entities are matched against existing graph vertices before creating new nodes. This ensures that "OpenAI" from one document and "OpenAI" from another map to the same canonical entity, building denser cross-document connections rather than parallel disconnected subgraphs.

Usage-based stability scoring. Facts that are frequently retrieved across queries accumulate stability scores, creating a signal for which knowledge is most actively relied upon. This metadata informs downstream retrieval prioritization.

D. Multi-Stage Query Pipeline.. When a query arrives, Vrin processes it through a multi-stage pipeline where each stage enriches the context for the next (see the query pipeline in Figure 1).

Stage 0: Adaptive Query Complexity Routing. Before entering the full pipeline, Vrin classifies each query's complexity as SIMPLE, MODERATE, or COMPLEX using structural signals—entity count, comparison markers, temporal references, causal language—without invoking an LLM (sub-1 ms overhead). Inspired by Adaptive-RAG Jeong et al. (2024), this classification determines retrieval depth, hop limits, and fan-out parameters, routing straightforward factual queries through a lightweight path while reserving the full multi-hop pipeline for genuinely complex questions. This avoids both the computational waste of over-retrieving for simple queries and the under-retrieval that occurs when complex queries are given insufficient depth.

Stage 0b: Knowledge-Aware Query Planning. Before decomposing a query into sub-questions, Vrin consults the knowledge graph's structural metadata: which entities exist, which communities they belong to, and what relationship patterns connect them. This alignment step ensures that the retrieval strategy targets entities and relationships that actually exist in the graph, rather than generating sub-questions from the query text alone. For a question about "underinvested layers in the AI agent stack," the planner identifies relevant knowledge clusters (agent memory research, evaluation methodology, knowledge infrastructure) and expands the entity set with candidates the raw query text would not have surfaced. The planning phase adds approximately 200 ms of overhead but prevents wasted retrieval iterations on entities absent from the graph.

Stage 1: Parallel Entity and Constraint Extraction. Two operations run in parallel. **Entity extraction** identifies the entities mentioned in the query, which become starting points for graph traversal. **Constraint identification** extracts five constraint types:

1. **Temporal**: years, quarters, date ranges
2. **Numerical**: comparisons, ranges, specific values
3. **Entity**: specific subjects or objects required
4. **Comparison**: requests to compare entities or metrics
5. **Aggregation**: totals, averages, percentages

These constraints actively filter downstream retrieval; a temporal constraint narrows graph traversal to facts within the specified window; a numerical constraint filters by stored numeric values.

Stage 2: Multi-Strategy Graph Traversal. Using extracted entities as seeds, Vrin performs multi-hop beam search:

- **Hop 0**: Rich entity matching (exact, case-insensitive, fuzzy)
- **Hops 1+**: Batched expansion with multiplicative confidence decay per hop
- **Hub detection**: High-degree entities receive restricted fan-out
- **Path confidence floor**: Expansion stops on low-confidence paths

The traversal produces entity chains, temporal chains, and causal chains. A cross-document synthesizer identifies entities appearing across multiple documents, detects temporal overlaps, and flags contradictions.

For complex queries classified as multi-hop or comparative involving two or more entities, Vrin generates three traversal strategies: *focused* (aggressive confidence decay, narrow expansion), *balanced* (default parameters), and *exploratory* (conservative decay, broad expansion), executing them in parallel. Results are merged via reciprocal rank fusion, boosting facts discovered independently by multiple strategies.

Personalized PageRank (PPR) Complementary Retrieval. Inspired by HippoRAG 2's Gutierrez et al. (2025) use of PPR over knowledge graphs, Vrin constructs an in-memory graph (via igraph) from retrieved facts and runs Personalized PageRank seeded on the query's extracted entities. Synonym

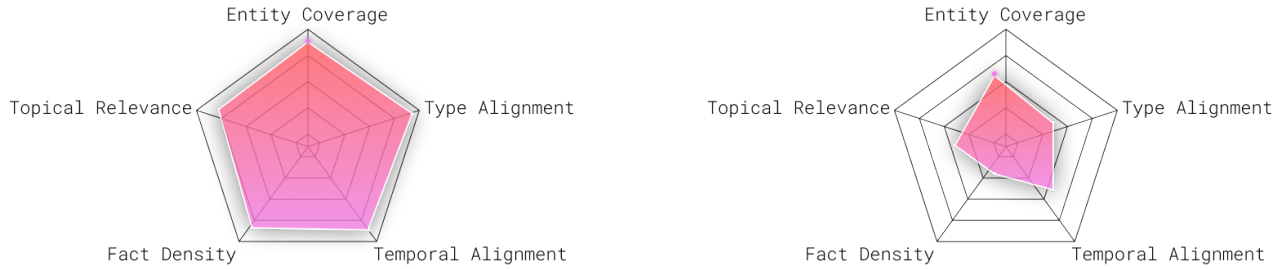

Fig. 2. Multi-dimensional retrieval confidence scoring. **Left**: A high-confidence retrieval where all five dimensions (entity coverage, type alignment, temporal alignment, fact density, topical relevance) score strongly, leading to direct generation. **Right**: A low-confidence retrieval with poor entity coverage and topical relevance, triggering either supplementary retrieval (ambiguous path) or adaptive bail-out.

edges based on substring and token overlap connect related entity nodes, enabling the PPR walk to discover facts that beam search alone misses—particularly facts connected through indirect or multi-step entity relationships. PPR results are merged with beam search results via reciprocal rank fusion, consistently improving recall on multi-hop queries without sacrificing precision.

Stage 3: Multi-Query Hybrid Search. An LLM selects one of two strategies per query: **Rephrase** (alternative phrasings for narrow queries) or **Decompose** (aspect-specific sub-queries for broad queries). Multiple variants are searched in parallel, each using hybrid BM25 keyword matching plus kNN vector similarity with score normalization and relevance filtering.

Stage 4: Retrieval Confidence Assessment. Before committing to LLM generation, Vrin evaluates retrieval quality across five dimensions: entity coverage, type alignment, temporal alignment, fact density, and topical relevance. Entity coverage measures what fraction of queried entities were found in the knowledge base. Topical relevance matches query terms against fact predicates and objects, detecting cases where an entity exists but the queried topic does not (for example, asking about a company's environmental record when only financial data is available).

The confidence assessment produces three outcomes, not two. When entity coverage is zero and no relevant facts or chunks were retrieved, the system short-circuits LLM generation entirely, returning a structured "insufficient information" response. This *adaptive bail-out* saves 1.5–5.5 seconds per null query while eliminating hallucinated answers about unknown entities. When entity coverage is high but topical relevance is zero, a *soft-miss bail-out* prevents the LLM from generating plausible-sounding but groundless responses.

When the confidence score falls in an intermediate range (0.25–0.55), the system enters an *ambiguous path* inspired by Corrective RAG Yan et al. (2024): rather than proceeding with uncertain evidence or bailing out prematurely, Vrin triggers supplementary retrieval using an exploratory strategy, merges the new evidence via reciprocal rank fusion, and re-scores confidence. This three-outcome design (proceed, supplement, bail out) reduces both false positives and false negatives compared to a binary threshold. Figure 2 illustrates the difference between high- and low-confidence retrievals across the five scoring dimensions.

Stage 5: Result Fusion and Reranking. Reciprocal Rank Fusion (RRF) Cormack et al. (2009) combines graph facts and vector chunks in a score-scale-independent manner. A cross-encoder reranker then re-scores candidates against the original query, producing significant precision improvements.

Stage 6: Iterative Reasoning with Quality Evaluation. For complex queries, Vrin decomposes the question into dependency-ordered sub-questions and performs targeted retrieval for each. Each iteration is evaluated: the system snapshots its state before retrieving, scores retrieval confidence after, and reverts the iteration if the new evidence degraded overall quality. This prevents noise accumulation from unproductive retrieval passes, a common failure mode in iterative retrieval systems where additional iterations add irrelevant facts that dilute the evidence pool.

Stage 7: Structured Context Preparation. Before the LLM generates a response, Vrin assembles a structured briefing fundamentally different from a list of text chunks. Facts are organized by entity and topic area rather than by source document, with an explicit coverage map telling the LLM how many knowledge clusters the evidence spans. Cross-document connections identified during graph traversal are stated as established insights, not left for the LLM to discover. The iterative reasoning chain (sub-questions, findings per step, confidence assessments) is injected as a structured reasoning path. Multi-hop facts carry provenance annotations indicating how many relationship steps they are from the original query entities.

This means the LLM synthesizes from organized understanding rather than searching through unstructured fragments. The context is reasoned over before the model sees it.

Stage 8: Streaming Generation. A configurable LLM (supporting OpenAI, Anthropic, Google, xAI) generates the response, receiving the structured briefing: topic-grouped graph facts with confidence scores, cross-document reasoning insights, the iterative reasoning chain, relevant text chunks with attribution, visual descriptions, and constraint information.

E. Enterprise Data Sovereignty.. Vrin supports three deployment modes: Vrin Cloud, Hybrid Cloud, and Private VPC. For enterprise customers, the knowledge graph, vector index, document storage, and embedding computation reside entirely within the customer's cloud account. Vrin's compute layer accesses customer data through time-limited, scoped credentials. The API key prefix transparently determines which infrastructure a request uses. Enterprise data never leaves the customer's cloud.

3. Implementation

Vrin is implemented as a serverless system on AWS: Lambda functions with FastAPI for streaming, Neptune for the knowledge graph, OpenSearch for hybrid search, Bedrock for embeddings (Cohere embed-english-v3, 1024 dimensions) and reranking (Cohere Rerank 3.5), S3 for documents, and DynamoDB for metadata. Cost-efficient model selection uses lightweight models for preprocessing and more capable models for generation.

Vrin also operates as a **Model Context Protocol (MCP) server**, enabling integration with Claude, ChatGPT, and custom AI agents; any MCP-compatible assistant can query Vrin's knowledge graph as a reasoning backend.

4. Experimental Evaluation

A. Methodology.. We evaluate on two complementary benchmarks: MultiHop-RAG Tang and Yang (2024), designed for cross-document multi-hop reasoning over news articles, and MuSiQue Trivedi et al. (2022), a multi-hop QA benchmark constructed through single-hop question composition to resist reasoning shortcuts. Together they test both accuracy on realistic multi-document queries and precision on compositionally complex questions. Following BetterBench guidelines Reuel et al. (2024), we use fixed-seed sampling (seed=42), report confidence intervals where applicable, and make all evaluation code open-source.*

For MultiHop-RAG, a three-stage evaluation pipeline ensures fair assessment: (1) direct substring matching, (2) LLM-based answer normalization to extract core answers from verbose responses, and (3) semantic pattern matching. The same pipeline evaluates both Vrin and baselines. For MuSiQue, we use standard SQuAD-style evaluation: Exact Match (EM) and Token F1, with normalization (lowercasing, article/punctuation removal). Vrin's verbose reasoning responses are post-processed via GPT-4o-mini to extract short factoid answers before scoring.

B. MultiHop-RAG Results.. MultiHop-RAG Tang and Yang (2024) consists of 2,556 queries requiring reasoning across 2–4 news articles. We evaluate on 384 stratified samples.

Vrin outperforms GPT 5.2 by **+16.2 percentage points** even when GPT is given the exact same evidence documents. This controls for retrieval quality; the gap is entirely attributable to how evidence is structured and reasoned over. Figure 3 visualizes this comparison.

The per-type breakdown (Table 2) reveals where structured reasoning matters most. **Inference queries**: both systems perform well (Vrin 99.2%, GPT 98.4%), as these are single-hop lookups where entity-centric extraction and oracle context both suffice. **Comparison queries (Vrin +15.5pp)**: Vrin's constraint solver explicitly identifies comparison operations and structures retrieval to find both sides; GPT must infer the comparison structure from raw text. **Temporal queries (Vrin +48.9pp)**: the largest gap. Temporal versioning and constraint extraction ensure the right facts from the right time periods are retrieved, while GPT struggles to reason over temporal relationships in unstructured evidence. **Null queries (Vrin 95.5%, GPT 100%)**: Vrin's retrieval confidence assessment with adaptive bail-out (Section 3, Stage 4) detects zero entity coverage and topical mismatches before LLM generation, correctly identifying 95.5% of queries about absent entities or topics. GPT achieves 100% because oracle context directly signals when information is absent, a condition that does not exist in production retrieval settings.

Table 1. MultiHop-RAG benchmark results.

System	Accuracy	95% CI
Vrin	**95.1%**	[90.5, 99.7]
GPT 5.2 (w/ evidence)	78.9%	[74.3, 83.5]
Multi-Meta RAG + GPT-4	63.0%	—
IRCoT + GPT-4	58.2%	—
Standard RAG + GPT-4	47.3%	—

Vrin and GPT 5.2: 384 stratified samples each (seed=42). GPT receives oracle evidence documents. Published baselines from Tang and Yang (2024).

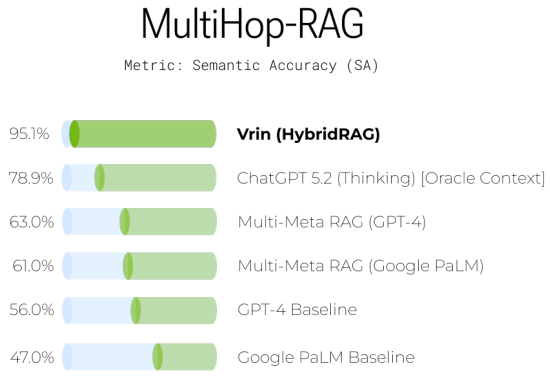

Fig. 3. MultiHop-RAG accuracy across systems. Vrin achieves 95.1% through retrieved context, outperforming GPT 5.2 (78.9%) with oracle context.

C. MuSiQue Results.. MuSiQue Trivedi et al. (2022) is a multi-hop question-answering benchmark constructed through single-hop question composition, ensuring that each question genuinely requires multi-step reasoning—single-hop shortcuts produce a 30-point F1 drop. Unlike MultiHop-RAG's accuracy metric, MuSiQue uses SQuAD-style Exact Match and Token F1, providing a complementary test of answer precision.

We evaluate on 300 questions sampled from the answerable validation split (seed=42, population 2,417), of which 299 completed successfully. The full supporting corpus of 4,848 Wikipedia paragraphs was ingested into Vrin's knowledge graph, yielding 39,412 stored facts after cross-paragraph deduplication. All pipeline features were active, including

*https://github.com/Vrin-cloud/vrin-benchmarks

Table 2. Accuracy by question type on MultiHop-RAG (384 samples).

Type	Vrin	GPT 5.2	n
Inference	**99.2%**	98.4%	123
Comparison	**94.6%**	79.1%	129
Temporal	**89.8%**	40.9%	88
Null	**95.5%**	100.0%	44

Stratified by question type. Vrin leads in all categories except null queries. Largest gaps: temporal (+48.9pp) and comparison (+15.5pp).

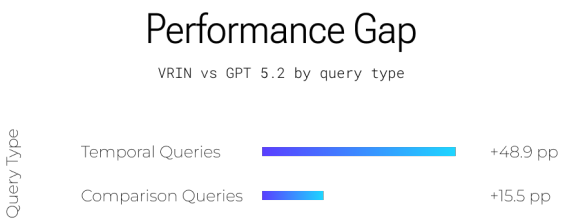

Fig. 4. Performance gap between Vrin and GPT 5.2 by query type. Temporal queries show the largest advantage (+48.9pp), followed by comparison queries (+15.5pp).

adaptive query complexity routing, PPR complementary retrieval, CRAG-inspired ambiguity handling, the iterative reasoning engine with query decomposition and per-iteration quality evaluation, knowledge consolidation with community detection, and knowledge-aware query planning.

Table 3. MuSiQue benchmark results (300 samples, seed=42).

System	Exact Match	Token F1
Vrin	**0.478**	**0.563**
HippoRAG 2	0.372	0.486
Standard RAG	—	0.457

Vrin: 299 completed out of 300 samples, full-corpus retrieval with knowledge consolidation. HippoRAG 2 and Standard RAG baselines from Gutierrez et al. (2025). Vrin leads on both Exact Match (+0.106) and Token F1 (+0.077).

Vrin achieves **0.478 Exact Match** and **0.563 Token F1**, surpassing HippoRAG 2's 0.372 EM (+0.106, a 28% relative improvement) and 0.486 F1 (+0.077, a 16% relative improvement). Vrin leads on both metrics. The improvement from the initial architecture (0.377 EM) to the current system (0.478 EM) was driven primarily by three additions: iterative reasoning with query decomposition (+24%), knowledge consolidation with community detection and deduplication (+2%), and knowledge-aware query planning that aligns retrieval with the graph's topology.

The per-complexity breakdown (Table 4) reveals that **simple queries** achieve the highest scores (EM=0.531, F1=0.613), with all tiers showing substantial improvement over the previous architecture. The iterative reasoning engine—which decomposes complex questions into dependency-ordered subquestions and performs targeted retrieval per identified gap—improves performance across all complexity tiers. Only 1.0%

Table 4. MuSiQue results by auto-routed query complexity.

Complexity	EM	Token F1	n
Complex	0.475	0.564	141
Moderate	0.430	0.544	100
Simple	**0.531**	**0.613**	49

Complexity assigned by structural classification (sub-1 ms, no LLM call). Simple queries achieve highest EM and F1, with all tiers benefiting from iterative reasoning.

of queries triggered the insufficient-coverage bail-out, down from ∼10% in early development—confirming the effectiveness of three-tier entity discovery (chunk titles, LLM extraction, query subject fallback) and adaptive vector thresholds.

D. Analysis.. The results across both benchmarks support three conclusions. First, **structured reasoning outperforms context stuffing**. The GPT 5.2 comparison on MultiHop-RAG controls for retrieval quality, isolating the value of structured processing. Second, **the gap is largest where structure matters most**: temporal queries show a 48.9 percentage point advantage for Vrin, and comparison queries show 15.5 percentage points. These are precisely the query types that require understanding the *structure* of the question (temporal constraints, comparison operations) rather than simply finding semantically similar text. Third, **the architecture generalizes across benchmarks**: Vrin achieves state-of-the-art or competitive results on both MultiHop-RAG (news article reasoning, accuracy metric) and MuSiQue (Wikipedia compositional reasoning, EM/F1 metrics), demonstrating that the hybrid knowledge graph approach is not overfit to a single evaluation paradigm.

E. Discussion: Oracle Context vs. Retrieved Context.. A critical distinction in RAG evaluation is the difference between *oracle context* (where the exact documents needed are placed in the LLM's context window) and *retrieved context*, where the system must find relevant information within a larger corpus containing both signal and noise. This distinction is often overlooked, yet it dominates real-world performance.

Recent work has quantified the gap. The RGB benchmark Chen et al. (2024) found that retrieval noise degrades LLM accuracy by 20–34 percentage points depending on model size. Meta's CRAG benchmark Yang et al. (2024) found that even the best commercial RAG system (Microsoft Copilot Pro) achieves only 62.6% accuracy on realistic retrieval tasks, with 16–25% hallucination rates across all tested systems. Google Research Joren et al. (2025) demonstrated that insufficient retrieved context increases error rates by 6.5× compared to having no context at all; RAG with bad retrieval is worse than no RAG.

In our evaluation, GPT 5.2 received the exact evidence documents for each query directly in its context window, an oracle setup where retrieval quality is perfect by construction. Vrin, by contrast, retrieved relevant facts and passages from the full ingested corpus of 609 news articles, which contains both relevant and irrelevant content for any given query. That Vrin achieves 95.1% through noisy retrieval while GPT 5.2 achieves only 78.9% with oracle context suggests the effective reasoning gap is substantially larger than the headline 16.2

percentage points.

This asymmetry reflects a broader reality: enterprise RAG systems operate on private, noisy, multi-source corpora where retrieval quality varies by query. No major enterprise RAG provider, including those valued at billions of dollars, competes on standardized public benchmarks. The industry has converged on custom evaluations against real enterprise data as more meaningful measures than leaderboard positions on academic benchmarks with artificial retrieval conditions. Vrin's benchmark methodology is designed with this in mind: the system ingests the full document corpus and retrieves under realistic conditions, then is compared against the strongest possible baseline: a frontier LLM with perfect context.

5. Production Readiness

Vrin operates as a production system. Expert-mode queries complete in under 20 seconds; simpler queries in 3–5 seconds. Null queries (questions about entities or topics absent from the knowledge base) are detected and answered in under 500 milliseconds through the adaptive bail-out system, eliminating unnecessary LLM calls. Responses stream in real-time via Server-Sent Events. The serverless architecture scales automatically with demand. All external calls include timeout, retry, and exponential backoff.

The system supports multiple LLM providers (OpenAI, Anthropic, Google, xAI) and operates as an MCP server for integration with Claude, ChatGPT, and custom agents. Current traction includes one finalized enterprise contract, two active pilots (including UC Davis Health), and five to seven additional enterprises in the pipeline, all inbound.

6. Vision: The 95% Unexplored

We believe the RAG industry has explored less than 5% of the available innovation space. The dominant focus has been on improving the *retrieval* subprocess: better embeddings, smarter reranking, larger context windows. Yet cognitive science has studied all five subprocesses for decades—perception, structuring, storage, organization, and retrieval each have established research traditions Kumaran et al. (2016); Collins and Loftus (1975); Fleming (2024). Four of those five, each with validated science behind them, remain largely unapplied in AI systems.

Vrin's five-subprocess framework reveals a vast surface area for future innovation:

Adaptive Retrieval. Vrin's confidence-based assessment system, inspired by DeepRAG's Guan et al. (2025) modeling of retrieval as a decision process and extended with CRAG-inspired Yan et al. (2024) supplementary retrieval, already makes three-outcome decisions: proceed, supplement, or bail out. Future versions will make even finer-grained adaptive decisions about which pipeline stages to invoke. Simple factual queries may need only graph traversal, while questions about general knowledge may not need retrieval at all.

Automatic Domain Specialization. Infrastructure exists for automatic specialization, learning domain expertise from query patterns and feedback. Future versions will detect that a knowledge base is finance-focused or healthcare-focused and adjust extraction, retrieval, and reasoning accordingly.

Iterative Reasoning. For queries requiring multi-step reasoning, Vrin implements iterative decomposition: each query is broken into dependency-ordered sub-questions, with targeted retrieval per identified gap, per-iteration quality evaluation (reverting unproductive iterations), and structured chain-of-thought injection. Combined with knowledge consolidation and knowledge-aware query planning, the full pipeline improved MuSiQue EM from 0.377 to 0.478 (+27%). Future versions will extend this to even deeper reasoning chains (5+ steps) with adaptive iteration budgets.

Knowledge Graph Pattern Detection and Model Specialization. Over time, usage patterns reveal which subgraphs and entity clusters are most frequently retrieved: specific teams repeatedly query the same financial entities, the same regulatory frameworks, the same product hierarchies. Vrin is building infrastructure to detect these patterns over months of usage and automatically create memory packs from the most heavily-accessed subgraphs. These memory packs then become the foundation for fine-tuning smaller, domain-specialized models. A model trained on a healthcare team's most-queried knowledge subgraph will outperform a general-purpose model on that team's queries while running at a fraction of the cost. This creates a virtuous cycle: structured knowledge in the graph enables precise pattern detection, pattern detection enables targeted memory pack creation, and memory packs enable efficient domain specialization per team and per concept.

The fundamental thesis is that AI systems will eventually be specialized like human employees, not through fine-tuning a single model, but through engineering the cognitive infrastructure surrounding it. Vrin is building that infrastructure.

7. Conclusion

The transition from retrieval to reasoning is not incremental; it is architectural. Vrin demonstrates that this transition is both possible and measurable: 95.1% on MultiHop-RAG versus 78.9% for GPT 5.2 with the same evidence documents, and 0.478 Exact Match and 0.563 Token F1 on MuSiQue versus HippoRAG 2's 0.372 EM and 0.486 F1, surpassing the state of the art on compositional multi-hop QA on both metrics by substantial margins.

These results are achieved not by using a more powerful language model, but by engineering the cognitive infrastructure around the model: entity-centric extraction with batch cross-chunk context, bi-temporal fact versioning, knowledge consolidation with community detection and deduplication, knowledge-aware query planning, constraint-aware retrieval, adaptive query complexity routing, multi-hop traversal with PPR complementary retrieval, iterative reasoning with per-step quality evaluation, hybrid search with rank fusion, three-outcome confidence assessment, and structured context preparation that organizes evidence by concept rather than by source.

The gap between retrieval and reasoning represents the largest opportunity in enterprise AI. Our benchmark evaluation code is open-source at https://github.com/Vrin-cloud/vrin-benchmarks.

ACKNOWLEDGMENTS. We thank the early enterprise partners and pilot users who provided feedback that shaped Vrin's architecture. Benchmark datasets are provided by the MultiHop-RAG and MuSiQue teams.

References

Anthropic (2024). Contextual retrieval. https://www.anthropic.com/news/contextual-retrieval.

Asai, A., Wu, Z., Wang, Y., Sil, A., and Hajishirzi, H. (2024). Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*.

Bakermans, J. J. W., Warren, J., Whittington, J. C. R., and Behrens, T. E. J. (2025). Constructing future behavior in the hippocampal formation through composition and replay. *Nature Neuroscience*, 28(5):1061–1072.

Chen, J., Lin, H., Han, X., and Sun, L. (2024). Benchmarking large language models in retrieval-augmented generation. *Proceedings of AAAI*.

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

Cormack, G. V., Clarke, C. L., and Buettcher, S. (2009). Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference*, pages 758–759.

Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., and Zhang, W. (2014). Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 601–610.

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Osazuwa Ness, R., and Larson, J. (2024). From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Fleming, S. M. (2024). Metacognition and confidence: A review and synthesis. *Annual Review of Psychology*, 75:241–268.

Guan, X., Zeng, J., Meng, F., Xin, C., Lu, Y., Lin, H., Han, X., Sun, L., and Zhou, J. (2025). DeepRAG: Thinking to retrieve step by step for large language models. *arXiv preprint arXiv:2502.01142*.

Gutierrez, B. J., Shu, Y., Gu, Y., Yasunaga, M., and Su, Y. (2024). HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*.

Gutierrez, B. J., Shu, Y., Sun, W., Gu, Y., and Su, Y. (2025). From RAG to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*.

Jeong, S., Baek, J., Cho, S., Hwang, S. J., and Park, J. C. (2024). Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 7036–7050.

Joren, H., Zhang, J., Ferng, C.-S., Juan, D.-C., Taly, A., and Rashtchian, C. (2025). Sufficient context: A new lens on retrieval augmented generation systems. *Proceedings of ICLR*.

Kumaran, D., Hassabis, D., and McClelland, J. L. (2016). What learning systems do intelligent agents need? Complementary learning systems theory updated. *Trends in Cognitive Sciences*, 20(7):512–534.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Li, D., Niu, Y., Ai, Y., Zou, X., Qi, B., and Liu, J. (2025a). T-GRAG: A dynamic GraphRAG framework for resolving temporal conflicts and redundancy in knowledge retrieval. *arXiv preprint arXiv:2508.01680*.

Li, F., Fang, P., Shi, Z., Khan, A., Wang, F., Wang, W., Zhang, X., and Cui, Y. (2025b). CoT-RAG: Integrating chain of thought and retrieval-augmented generation to enhance reasoning in large language models. *Findings of EMNLP*.

Neelakantan, A., Xu, T., Puri, R., Radford, A., Han, J. M., Tworek, J., Yuan, Q., Tezak, N., Kim, J. W., Hallacy, C., et al. (2022). Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*.

Nogueira, R. and Cho, K. (2019). Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.

Reuel, A., Hardy, A., Smith, C., Lamparth, M., Hardy, M., and Kochenderfer, M. J. (2024). BetterBench: Assessing AI benchmarks, uncovering issues, and establishing best practices. In *Advances in Neural Information Processing Systems*. https://betterbench.stanford.edu/.

Tang, Y. and Yang, Y. (2024). MultiHop-RAG: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*.

Traag, V. A., Waltman, L., and van Eck, N. J. (2019). From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1):5233.

Trivedi, H., Balasubramanian, N., Khot, T., and Sabharwal, A. (2022). MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Uytiepo, M., Zhu, Y., Bushong, E., et al. (2025). Synaptic architecture of a memory engram in the mouse hippocampus. *Science*, 387(6740):eado8316.

Wang, Z., Peng, B., Tu, H., and Li, X. (2025). Entity similarity RAG: Enhancing LLM answers with precise knowledge graph retrieval. In *Proceedings of ICONIP*. Springer.

Webb, T., Mondal, S. S., and Momennejad, I. (2025). A brain-inspired agentic architecture to improve planning with LLMs. *Nature Communications*, 16:8633.

Whittington, J. C. R., Warren, J., and Behrens, T. E. J. (2022). Relating transformers to models and neural representations of the hippocampal formation. In *International Conference on Learning Representations*.

Yan, S.-Q., Gu, J.-C., Zhu, Y., and Ling, Z.-H. (2024). Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*.

Yang, X., Sun, K., Xin, H., Sun, Y., Bhalla, N., Chen, X., Cai, S., Dang, H., Sunkara, K., Jiang, Y., et al. (2024). CRAG – comprehensive RAG benchmark. *arXiv preprint arXiv:2406.04744*.