A Feedback-Driven and Self-Optimizing RAG Framework for Enterprise Knowledge Infrastructure

Han Han, Funing Wang, Jing Cai, Xiaokang Chen, Longbo He, Hao Sun, Qiuyi Yu

Beijing Zhongshuruizhi Co., Ltd.

Abstract

Enterprise knowledge platforms often struggle with static frameworks, costly maintenance, and limited adaptability to evolving tasks. We present *5D-RAG*, a modular, feedback-driven RAG framework that supports *real-time adaptation* via multi-agent collaboration and online learning. The framework decomposes the retrieval process into five specialized optimizers, enabling fine-grained control and continuous self-improvement through user and system feedback. This agent-based decomposition facilitates robust generalization across dynamic knowledge updates and heterogeneous enterprise scenarios. 5*D-RAG* integrates full-lifecycle knowledge management and supports plug-and-play extension while ensuring *low-latency, high-accuracy performance* under complex constraints. Empirical results from real-world deployments demonstrate its *scalability*, *efficiency*, *and domain sensitivity*, reducing manual intervention and improving response quality in enterprise-level applications.

1. Introduction

Enterprise knowledge management systems face persistent challenges in three core areas: limited adaptability to diverse scenarios, delayed integration of user feedback, and suboptimal retrieval accuracy-issues exacerbated by the scale and heterogeneity of unstructured enterprise data. While conventional Retrieval-Augmented Generation (RAG) frameworks have demonstrated utility in static environments, they fall short in dynamic knowledge flows and real-time operational demands.

To address these limitations, we propose *5D-RAG*, a self-optimizing, feedback-driven RAG framework built upon a multi-agent collaborative framework with online learning capabilities. At its core lies a *Five-Dimensional Optimizer Framework*, comprising five specialized modules—*Rewriting Optimizer*, *Classification Optimizer*, *TR² Optimizer (Tag-Refine Retrieval Optimizer)*, *Retrieval Optimizer*, and *Summarized Optimizer*—each targeting a distinct functional layer of the RAG pipeline. These modules form a closed-loop optimization system that incrementally refines both retrieval and response through iterative feedback incorporation.

We have deployed *5D-RAG* into the *YuSi Knowledge Middleware Platform*, an enterprise-grade infrastructure designed to support the full lifecycle of unstructured knowledge, from extraction and fusion to refinement and delivery. The platform is modular, horizontally scalable, and has been successfully deployed in sectors including energy, manufacturing, telecommunications, and defense. It provides foundational support for knowledge-centric engineering and LLM application development in enterprise environments.

2. Technical Approach—5D-RAG

To enable continuous adaptation to large-scale, evolving knowledge environments, *5D-RAG* introduces an online learning paradigm that enhances traditional RAG pipelines across five dimensions.

Rewriting Optimizer Reformulates raw user queries into semantically enhanced variants to improve intent recognition, retrieval routing, and overall answer fidelity.

Classification Optimizer Implements a fine-grained scenario classifier that routes queries to corresponding sub-indexes via a *partition-first, retrieval-later* strategy, balancing retrieval accuracy, system efficiency, and data compliance.

Tag-Refine Retrieval (TR²) Optimizer Applies an *extract-then-match* strategy to generate dynamic tag sets by extracting explicit entities and implicit semantic labels during indexing. At retrieval, query tags are matched with chunk tags, and tag-based relevance is fused with vector similarity to enhance semantic alignment and recall robustness.

Retrieval Optimizer Uses online learning from user feedback, multi-armed bandit decision logic, and semantic distribution modeling to adapt retrieval parameters in real across domains.

Summarized Optimizer Leverages intelligent routing, multi-dimensional feature-aware parameter tuning, and feedback-driven prompt engineering to enable end-to-end adaptive optimization of model selection, generation control, and response quality.

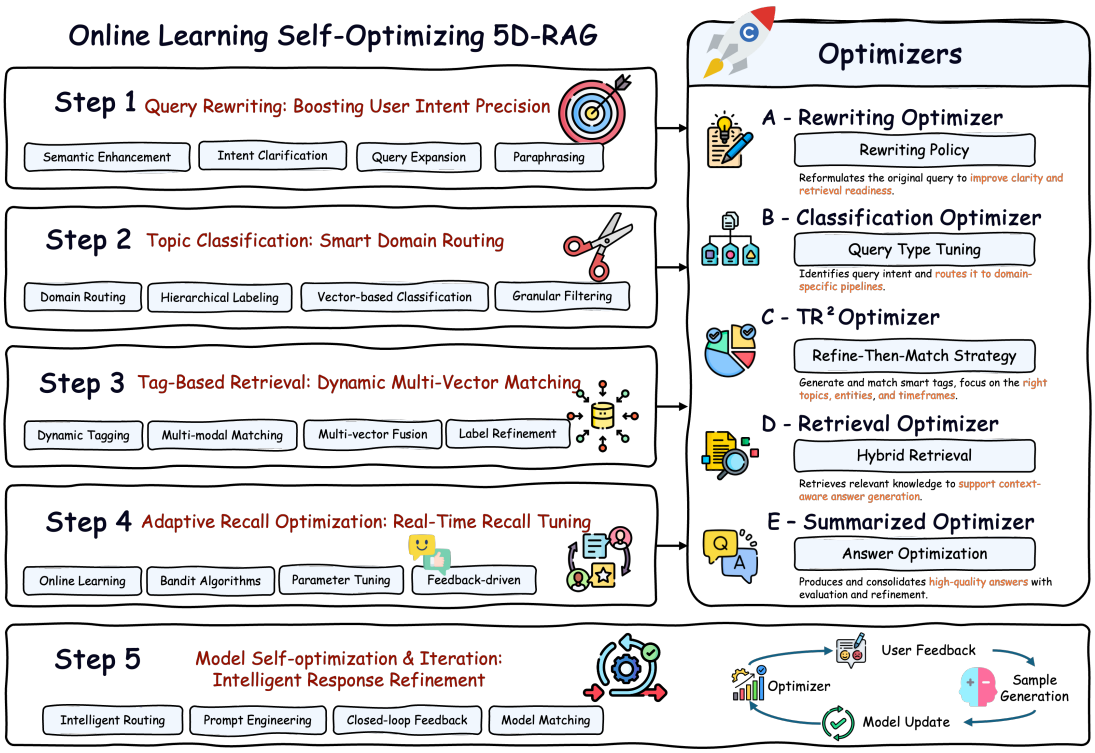

Fig.1 The 5D-RAG framework employs modular, feedback-driven optimizers to support continual learning and knowledge refinement across the retrieval-to-generation workflow.

2.1. Scenario and Motivation

In Retrieval-Augmented Generation (RAG) systems, prompt quality critically affects both retrieval accuracy and generation performance. However, raw user queries are often colloquial, ambiguous, or vague. Directly using such queries for retrieval can lead to semantic misalignment and suboptimal results.

For example, the query "What recent changes have occurred in company products?" contains a vague temporal reference ("recent") and lacks specificity about product categories, resulting in an unfocused retrieval scope and failure to meet user needs precisely.

To mitigate this, the *Rewriting Optimizer* reformulates user queries into semantically enriched prompts, enhancing intent interpretation and improving the retrieval path planning process.

As business knowledge expands rapidly, traditional RAG systems encode all document chunks uniformly into a single vector store, encountering semantic confusion, scalability bottlenecks, and compliance challenges. Domain-specific terminologies interfere in the shared embedding space, causing retrieval drift; large index sizes induce significant latency; and multi-tenant data isolation increases deployment complexity. The *Classification Optimizer* adopts a *partition-first, retrieve-later* strategy by routing knowledge chunks into thematic sub-indexes at indexing time and predicting query categories to direct retrieval, thus balancing accuracy, efficiency, and compliance.

Conventional recall strategies are also prone to errors stemming from entity ambiguity and temporal misalignment. *For instance, "Apple" may ambiguously denote a company or a fruit, and superseded tax rates may still be retrieved incorrectly.* The *TR² Optimizer* introduces dynamic tagging to address this: it generates explicit and implicit tags—including temporal, entity, and intent information—and refines retrieval through tag matching and temporal filtering, thereby reduce semantic drift and enhancing retrieval relevance.

In dynamic business scenarios, fixed-parameter recall strategies often underperform due to the lack of adaptability. The *Retrieval Optimizer* addresses this by leveraging both explicit feedback and implicit user interactions. It employs online learning, multi-armed bandit algorithms, and semantic distribution analysis to iteratively adjust retrieval parameters. This supports real-time optimization and ensures robust performance across heterogeneous retrieval contexts.

Finally, the *Summarized Optimizer* improves response accuracy in complex and domain-sensitive scenarios. By incorporating intelligent routing, feature-aware prompt engineering, and runtime monitoring, it dynamically allocates model resources and tunes parameters based on feedback. This enables high-quality answer generation even under tight computational budgets and evolving query complexities.

2.2. Rewriting Optimizer

2.2.1. OPTIMIZATION OBJECTIVE

The *Rewriting Optimizer* aims to transform raw user queries (Q) into clearer, more structured reformulations (Q') via semantic-enhanced prompt rewriting. This transformation improves intent comprehension, retrieval path planning efficiency, and answer matching accuracy.

The optimization focuses on three key directions to ensure a deterministic and semantically enriched conversion from Q to Q':

1 *Efficient Utilization of Structured Metadata.* Leveraging inherent document features—such as title hierarchy, section logic, entity relations, and temporal annotations—as retrieval cues, enabling rapid localization of target content while minimizing irrelevant noise.

2 *Dynamic and Precise Retrieval Control.* Guiding the model with prompts to flexibly select retrieval granularity based on query complexity (e.g., simple lookup vs. multi-hop reasoning), thereby avoiding inefficient search operations.

3 *Precise User Intent Locking.* Converting colloquial or vague inputs (*e.g., "recent updates"*) into explicit instructions (*e.g., "product updates from July 2024 to July 2025"*), narrowing the retrieval scope and substantially improving hit accuracy.

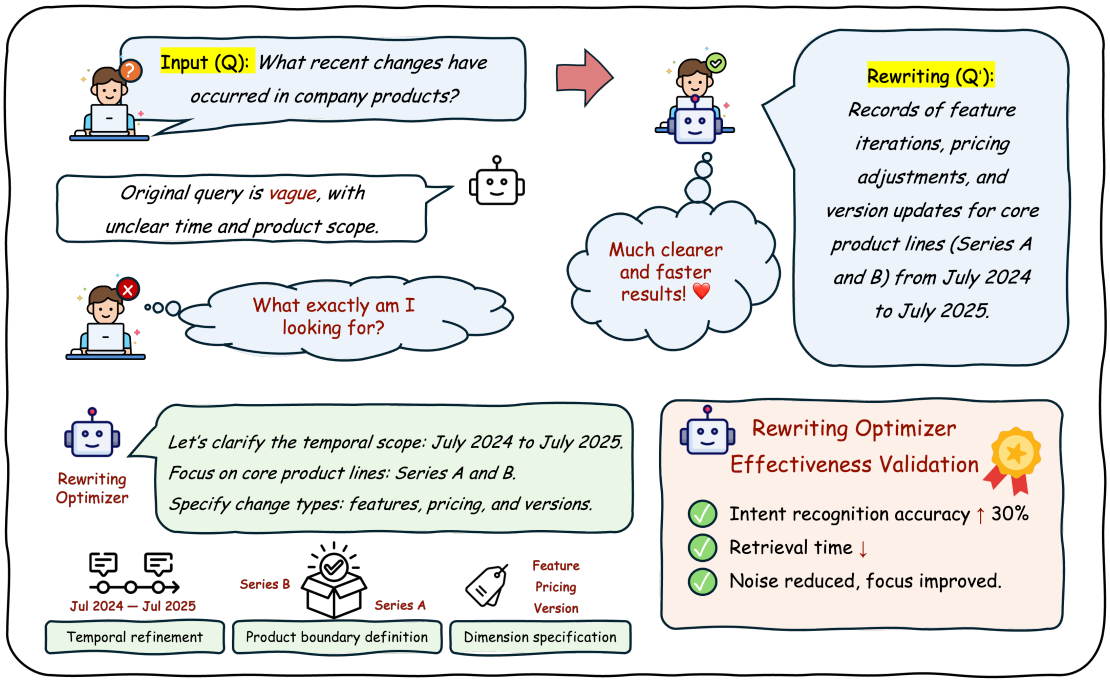

Fig.2 Case study of rewriting optimization for the query "*What recent changes have occurred in company products?*"

2.2.2. OPTIMIZATION PRINCIPLES

The *Rewriting Optimizer* implements five core technical pathways to convert raw queries into semantically explicit formulations:

1 *Injection of Structured Metadata.* Extract document structural elements such as titles, section numbers, entity attributes, and timestamps from original prompts, embedding them as standardized tags. This enriches semantic discriminability and enables rapid content boundary localization.

2 *Multi-Granularity Index Invocation.* Construct multi-level indices based on document structure (paragraphs, sections, titles). Dynamically adjust retrieval granularity via prompt guidance: simple queries use title-level indices; complex queries invoke paragraph-level; cross-topic queries employ hybrid section-title indices—balancing precision and efficiency.

3 *Structured Retrieval Path Planning.* For multi-hop or complex queries, integrate chain-of-thought reasoning to rewrite inputs as sequential retrieval steps. For instance, "*AI projects and responsible personnel for a client*" is decomposed into retrieving "*client-project*" associations, filtering projects tagged with "*AI technology*" then extracting project leads. This ensures logical and efficient retrieval.

4 *Fine-Grained Temporal Parsing.* Normalize temporal expressions in user inputs, converting vague references (*"last year"*) into exact time ranges (*"Jan 1, 2024 – Dec 31, 2024"*). This enhances temporal awareness and mitigates retrieval errors from unclear time boundaries.

5 *Domain Boundary Delimitation.* Add domain-specific tags (*e.g., "[Domain: Clinical Medicine]"*) to clarify query scope and filter cross-domain noise. For example, *"hypertension treatment protocols"* is tagged "*[Domain: Clinical Medicine] [Condition: Hypertension]*" to ensure domain-relevant retrieval.

2.2.3. TECHNICAL ADVANTAGES

1 *Improved Intent Parsing.* Structured prompts raise recognition accuracy of temporal, scope, and target dimensions by over 30%.

2 *Enhanced Retrieval Efficiency.* Dynamic granularity indexing combined with structured retrieval paths significantly reduces redundant recalls and lowers computational overhead.

3 *Better Result Quality Control.* Domain constraints and semantic prompt enrichment decrease false positives, boosting relevance and user satisfaction.

4 *Dynamic Evaluation and Continuous Optimization.* Using metrics such as Hit@K, NDCG (Normalized Discounted Cumulative Gain), MRR (*Table 1* below) alongside user feedback, an iterative improvement framework is established.

Table 1. An Evaluation Metric for Rewriting Optimization

METRIC TYPE	METRIC NAME	DESCRIPTION
ACCURANCY	Hit@K	Proportion of relevant items in the top K results, measuring retrieval accuracy
	NDCG	Measures ranking quality by prioritizing relevant results higher
EFFICIENCY	Retrieval Latency	Time from user input to result delivery
	Computational Cost	Hardware resource consumption
DIVERSITY	MRR	Measures efficiency in retrieving the first relevant item
	Shannon Entropy	Reflects diversity in result sources

To ensure continued effectiveness in dynamic environments, we adopt an *iterative optimization design* that supports real-time adaptability and long-term improvement. We employ *A/B testing* to systematically compare different prompt strategies, such as metadata injection versus standard prompts and varying levels of retrieval granularity, to identify optimal configurations. *User feedback integration* is conducted through regular collection of satisfaction ratings and improvement suggestions, which are used to refine model behaviors over time. Furthermore, *context-aware adaptation* allows parameter tuning based on enterprise-specific scenarios—for example, adjusting for frequent multi-hop queries or time-sensitive retrieval in dynamic business domains.

2.3. Classification Optimizer

2.3.1. OPTIMIZATION OBJECTIVE

In a typical RAG pipeline, all document chunks are uniformly encoded into a single vector store. As domain knowledge expands rapidly, this monolithic design suffers from three major issues:

- *Semantic Confusion*: Terminological overlap across domains—such as finance, law, and healthcare—leads to top-k result drift due to vector space interference.
- *Scalability Bottlenecks*: Index bloating and increased latency emerge as the number of chunks scales to billions, with ANN (Approximate Nearest Neighbor) search often exceeding 800 ms per query.
- *Access Control & Compliance*: Data across tenants or security levels require physical separation.

Traditional solutions entail deploying multiple RAG pipelines, resulting in significant infrastructure overhead.

The *Classification Optimizer* is designed to address these core pain points—semantic confusion, retrieval latency, and access isolation—by introducing a taxonomy-aware, multi-topic vector routing mechanism. This enables high-precision, low-latency, and compliance-ready retrieval.

2.3.2. OPTIMIZATION PRINCIPLES

The *Classification Optimizer* follows a *partition-first, retrieve-later* strategy, introducing taxonomy-aware retrieval that routes chunks to different vector sub-indices based on thematic classification during ingestion. During querying, the user question is first classified to determine the most relevant sub-index, thereby improving retrieval accuracy, latency, and compliance.

2.3.1.1 Offline Pipeline: Chunk Classification & Routing

Chunks are no longer stored in a single large-scale table. Instead, a lightweight *semantic router* assigns a topic label to each chunk, which is then stored in a physically isolated vector sub-index.

1 *Taxonomy Design.* A three-level taxonomy (*Domain → Topic → Subtopic*) is dynamically constructed by combining document metadata (*e.g., department, product line*), pre-trained topic models (embedding + fine-tuned K-means), and human validation to generate high-quality topic labels.

2 *Chunk Routing.* Document slices—produced via semantic or sliding window chunking—are passed through a routing model (a 12M-parameter distilled MPNet), which outputs a topic probability distribution p(topic|chunk). If the top probability exceeds 0.65, the chunk is routed to the corresponding sub-index; otherwise, it is sent to a fallback index. Each sub-index is constructed independently using HNSW (Hierarchical Navigable Small World) graphs, supporting encrypted storage and modular scaling.

2.3.1.2 Online Pipeline: Query Classification & Targeted Retrieval

User queries are classified by the same router, and ANN search is only executed on the matching sub-index.

1 *Query Topic Prediction*
The query q is processed to predict its topic. If the top probability is below 0.4, a multi-topic fallback is triggered: the top-3 predicted sub-indices are queried in parallel, with results merged during reranking to enhance robustness.

2 *Sub-index Retrieval & Fusion*
ANN search is performed only on the relevant sub-index (with *ef_search=128*). Final ranking combines vector similarity and topic match probability:

$$score = \alpha \cdot \frac{1}{distance} + \beta \cdot \rho\left(\frac{topic}{q}\right)$$

Empirical results show optimal MRR@10 when $\alpha=0.8$ and $\beta=0.2$.

2.3.3. TECHNICAL ADVANTAGES

1 *Significant Semantic Precision Gains.* Taxonomy-aware routing effectively reduces cross-domain

interference. On a financial QA benchmark, Hit@5 improved from 71.3% to 83.7%, while false positives were nearly halved (–47%).

2 *Drastically Lower Latency.* With average sub-index sizes reduced to 1/12 of the original, ANN search latency dropped from 800ms to 95ms, delivering a substantial speed-up.

3 *Elastic Resource Utilization.* Each sub-index can be independently scaled. For low-traffic topics, the number of replicas can be reduced, saving up to 35% of GPU memory consumption.

4 *Built-in Compliance & Security*. Physically isolated sub-indices naturally support multi-tenant and classified data scenarios, without requiring redundant encryption or system duplication.

2.3.4. PERFORMANCE METRICS

Table 2. Performance Comparison of Classification Optimization

STRATEGY	HIT@5	LATENCY(MS)	INDEX SIZE(GB)	GPU MEMORY(GB)
BASELINE	71.3%	800	95	24
+RERANK	79.1%	1200	95	30
TAXONOMY-AWARE	83.7%	95	8x12=98（split）	18

Baseline: Unified vector index without reranking; + Rerank: Adds ColBERT reranking layer; Taxonomy-Aware: Classification-based vector rerouting with rerank fusion

2.4. TR² Optimizer

2.4.1. OPTIMIZATION OBJECTIVE

In high-stakes scenarios such as enterprise Q&A, financial analytics, and regulatory auditing, traditional vector-based retrieval systems often suffer from structural issues like semantic drift, entity ambiguity, and temporal mismatch. These errors severely undermine retrieval accuracy and downstream utility. The *TR² (Tag-Refine Retrieval) Optimizer* aims to build an interpretable, temporally-aware, and semantically-enhanced tag-driven retrieval mechanism to significantly improve recall precision and semantic alignment.

In industry applications, user queries often contain explicit temporal constraints—e.g., "*Tesla's Q4 2023 revenue*" or "*COVID oral drugs approved before Sep 2022*." Traditional RAG pipelines struggle to extract and enforce such time conditions, resulting in three types of critical errors:

- Temporal Drift: Data from 2024 used to answer a 2022-specific question.
- Temporal Omission: Outdated policies (e.g., from 2021) returned due to a lack of filtering.
- Temporal Conflict: Mixed versions (2020, 2022, 2023) of a policy are shown in a single answer.

2.4.2. OPTIMIZATION PRINCIPLES

TR² Optimizer adopts a *refine-then-match* strategy to enhance the retrieval flow through tagging. It introduces dual-path tag generation, query tag parsing, tag-matching reranking, and temporal filtering with weighting, to effectively address semantic drift, entity confusion, and temporal inconsistency.

2.4.2.1. General Tag Optimization Mechanism

1 Dual-Path Tag Generation

- Slow Path (LLM-based)
  A 7B instruction-tuned language model is prompted to extract both explicit entities and implicit semantics (e.g., intent, emotion).

  Prompt: "Please generate up to 8 tags for the following text. Return them as a JSON array."

  Example output: ["liability determination", "2025-03-15", "insurance claim", "Article 76 of traffic law"]

- NER
  A 3-layer Bi-LSTM+CRF model performs Named Entity Recognition (NER) in real time, focusing on standard entities like names, organizations, and locations.

- Tag Merging Strategy
  Results from both paths are de-duplicated and merged via confidence-weighted voting. When overlap occurs, the higher-confidence tag is retained; otherwise, all tags are preserved.

2 Tag Storage

Tags are stored alongside vectors in a key-value format.

Tags are embedded using *Sentence-T5-mini* for high-efficiency matching.

3 Query Tag Parsing & Retrieval Reranking

a) A user query q is parsed into a tag set Q_{tags} using the same dual-path pipeline.

b) For each retrieved chunk cc, tag similarity is computed:

$$\text{tag\_sim}(q, c) = \max_{t \in Q\_tags} \cos\big(\text{tag\_emb}(t), \text{tag\_emb}(t_c)\big)$$

c) The final retrieval score is a weighted combination:

$$\text{hybrid} = \alpha \cdot \text{vec\_sim} + \beta \cdot \text{tag\_sim} + \gamma \cdot \text{exact\_match}$$

Optimal MRR@10 is observed with α=0.55, β=0.35, γ=0.10.

2.4.2.2. Temporal Awareness Optimization

TR² injects temporal awareness into both document chunks and user queries, resolving time-related errors during retrieval.

1 Offline Phase: Timeline Construction & Time Indexing

a) Time Extraction & Normalization

Rule-based Layer: Regex and dictionaries extract absolute (e.g., "2023-12-31") and relative times (e.g., "last year"), along with event-anchored dates (e.g., "IPO date", "FDA approval day").

Model-based Layer: LLMs annotate sequences with time-related entities using tags like B-Time, I-Time, B-Event, etc.

Standardization: All times normalized to ISO-8601 format. Relative times are resolved using the document's publication date as a fallback anchor.

b) Chunk-Level Time Attribution

Chunking: Sliding window of 512 tokens (64-token overlap), preserving semantic integrity for tables/lists.

Time Association: LLM determines each chunk's "primary time"—the point at which the content is most applicable.

Prompt: "Read the following passage and determine the most applicable year and quarter. If unavailable, return 'NA.'"

c) Time fields are stored as Unix timestamps; a secondary B+ tree time index is built alongside the HNSW semantic index.

2 Retrieval Phase: Time-Constrained Filtering & Refill

a) Time Span Parsing

User queries with temporal constraints (e.g., "sales data for Q2 2023") are parsed into time intervals TimeSpan.

Fallback: If vague terms like "recent" or "latest" are used, a default range (e.g., last 180 days) is applied.

b) Two-Stage Retrieval

Stage A: Retrieve top-k results via semantic similarity

Stage B: Apply time filters:

$$\text{start} \leq t_{end} \text{ and end} \geq t_{start}$$

c) Refill Strategy

If the filtered results are fewer than *k*, refill by selecting semantically similar documents with minimal time distance from the query period.

d) Time-Aware Reranking

Re-ranker: MiniLM cross-encoder, outputs semantic relevance s_{sem}

Temporal weight:

$$w_{time} = 1.0 - \min\left(\frac{\left|t_{mid} - t_{query\_mid}\right|}{365},\ 1.0\right)$$

Final score:

$$s_{final} = 0.7 \cdot s_{sem} + 0.3 \cdot w_{time}$$

2.4.3. TECHNICAL ADVANTAGES

Table 3. Tag Optimization Ablation Study

STRATEGY	Hit@5	FALSE OMISSION RATE	AVG TAGS PER CHUNK	EXTRA LATENCY(ms)
PURE VECTOR	72.4%	15.6%	0	0
+ KEYWORDS	76.1%	12.3%	4	2
TR² (OURS)	85.7%	6.1%	5.3	5

Additional Benefits:

1 *High Explainability.* Returned answers include a tag cloud, helping users understand "why it was retrieved."

2 *Tag-level Access Control.* Supports fine-grained permission strategies based on tags, including redaction and filtering.

3 *Low-Cost Incremental Update.* New tags only require appending new tag embeddings—no need to recompute the entire vector space.

2.5. Retrieval Optimizer

2.5.1. OPTIMIZATION OBJECTIVE

The retrieval module directly determines both the information coverage and response efficiency of a search system. Traditional static configurations—such as a fixed number of retrieved candidates or fixed similarity thresholds—struggle to adapt to dynamic changes in user behavior, data distribution, and evolving business goals. This often leads to reduced recall precision and delayed responsiveness to new trends. In scenarios where system performance is sensitive and user feedback fluctuates rapidly, the *Retrieval Optimizer* must support real-time, self-adaptive mechanisms that strike an optimal balance between robustness and sensitivity.

2.5.2. OPTIMIZATION PRINCIPLES

The core of retrieval optimization lies in a *multi-layered dynamic decision system* that combines online learning with multi-dimensional analytics to achieve parameter auto-tuning and enhanced retrieval precision. The *Retrieval Optimizer* introduces multi-stage online learning mechanisms that leverage user feedback and document distribution patterns to adjust parameters dynamically.

1 Online Learning Driven by User Feedback

The system continuously collects explicit user feedback (e.g., relevance scores) and implicit behavioral signals (e.g., click-throughs, dwell time). Using initial parameters such as *Top-K* (number of retrieved items) and *score threshold*, it applies a sliding-window weighted update strategy based on a *Sliding Window Multi-Armed Bandit (SW-MAB)* framework. An *Exponentially Weighted Moving Average (EWMA)* is used to smooth historical feedback, reduce the influence of outliers, and emphasize recent interactions, thereby improving sensitivity to emerging trends.

2 Explore–Exploit Trade-off Strategy

Within the parameter search space, the system dynamically chooses strategies using a *Multi-Armed Bandit (MAB)*model. During the exploration phase, parameter perturbations are introduced to avoid local optima; during exploitation, the system prefers parameter settings with historically strong performance. This iterative balance enables the system to converge toward a near-optimal global configuration that adapts across domains.

3 Collaborative Multi-Document Analysis

Semantic consistency across retrieved content is enhanced through document-level metrics such as the *Document Distribution Index (DDI)* and *Local Focus Ratio (LFR)*. When the DDI indicates content is concentrated in a few documents, the system switches to a long-document retrieval strategy. LFR is used in conjunction with the *Topic Coherence Score (TCS)* to assess semantic coherence across passages, mitigating hallucinations and reducing recall noise.

2.5.3. TECHNICAL ADVANTAGES

1 *Enhanced Adaptivity.* Real-time parameter tuning based on user feedback allows the system to automatically adjust to shifts in user behavior and business needs, reducing reliance on manual configurations.

2 *Robustness and Agility.* The combination of EWMA smoothing and the explore–exploit mechanism ensures both resistance to noisy feedback and avoidance of local optima, maintaining high-quality retrieval performance under uncertainty.

3 *High Semantic Precision.* The joint use of DDI, LFR, and TCS improves the accuracy of semantic consistency evaluation across multiple documents, reducing irrelevant or semantically incoherent results.

4 *Strong Generalization.* Compatible with various data domains and application targets, particularly effective in feedback-rich environments like recommendation systems and search engines.

2.5.4. PERFORMANCE METRICS

Table 4. Performance Metrics of Retrieval Optimizer

METRIC TYPE	OBSERVED PERFORMANCE
RECALL QUALITY	On standard datasets (MS MARCO, Natural Questions, BEIR, SQuAD 2.0): Top-K recall accuracy improved by 13–21% NDCG@10 increased by an average of 0.12
RESPONSE LATENCY	Parameter update latency kept under 50ms, supporting real-time interaction
STABILITY	EWMA smoothing reduced performance fluctuation due to feedback volatility by 30%; model convergence speed increased by 40%
SEMANTIC CONSISTENCY	TCS-guided optimization reduced cross-passage topic conflicts by 22%; hallucination risk decreased by 15%

2.6. Summarized Optimizer

2.6.1. OPTIMIZATION OBJECTIVE

In RAG systems, user questions are highly diverse in complexity, domain specificity, and contextual interaction. At the same time, LLMs, which power the generation component, are computationally expensive and often volatile in behavior. Traditional fixed model configurations and static parameter settings cannot effectively handle this dynamic complexity. In high-complexity queries or under resource constraints, these limitations often result in degraded answer quality and increased response latency.

The Summarized Optimizer is designed to build an intelligent question-answering optimization mechanism that improves robustness, adaptability, and responsiveness under challenging conditions. It achieves this by aligning question complexity with model capability, dynamically tuning generation parameters, and refining prompts through feedback-driven mechanisms.

2.6.2. OPTIMIZATION PRINCIPLES

Summarized Optimizer is built on a multi-layer intelligent decision and adaptive adjustment system that integrates smart routing, parameter self-tuning, and feedback-driven prompt engineering.

1 Intelligent Routing Decision Engine

The system performs real-time analysis of both the semantic characteristics of the question and the current status of available model resources. It constructs a multidimensional feature profile that includes syntactic complexity, semantic entropy, and domain classification.

- Syntactic complexity is assessed using dependency tree depth (≥4 denotes complex sentence structures).
- Question entropy is calculated via TF-IDF metrics.
- A hierarchical BERT classifier predicts domain labels at three levels of granularity.

Based on these features, the decision engine references a *Model Configuration Registry* (containing model size, domain affinity, routing rules, etc.) and computes a dynamic compatibility score using the following weighted function:

$$\text{Score}(M_i) = \alpha \cdot C(M_i, Q) + \beta \cdot D(M_i, Q) + \gamma \cdot R(M_i) + \delta \cdot L(M_i)$$

Where:

- Cognitive Capability Match:

$$C(M_i, Q) = \frac{P(M_i)}{P_{\max}} \times \text{sigmoid}\big(\kappa \times \text{Complexity}(Q)\big)$$

  $P(M_i)$: Parameter count of model M_i

  $P_{\max}$: Maximum parameter count among all models

  κ: Complexity sensitivity factor

- Domain Adaptation Score:

$$D(M_i, Q) = \sum_{j=1}^{n} w_j \times \cos\left(\boldsymbol{V_j}(M_i), \boldsymbol{V_j}(Q)\right)$$

  n: Number of domain levels

w_j: Weight of domain level jj

$V_j(M_i)$, $V_j(Q)$ Domain vectors of model and query respectively

- Resource Load $R(M_i)$ and Latency $L(M_i)$ are also incorporated to optimize final routing decisions.

2 Data-Aware Parameter Optimizer

The system builds a knowledge graph profile of the document corpus based on multiple document-level features:

- Text structure: average sentence/paragraph length, punctuation density
- Linguistic traits: core noun ratio, passive voice frequency
- Informational value: entropy, topic dispersion
- Semantic complexity: entity density, coreference resolution ratio

These features are used to dynamically tune key RAG parameters:

- Retrieval Top-K
- Semantic similarity threshold
- Context window size
- Generation temperature

3 Feedback-Driven Prompt Optimization System

A closed-loop feedback pipeline continuously improves prompt design:

- User feedback is categorized into 12 predefined types (e.g., redundancy, misunderstanding) using a fine-tuned DeBERTa classifier.
- For novel issue types, a language model summarizes key failure features and generates new prompt templates, which are stored in a Prompt Knowledge Base.
- Upon matching a similar query in the future, the system auto-injects the optimal prompt to guide generation accuracy.
- The entire workflow is automated, low-latency, and self-evolving.

2.6.3. TECHNICAL ADVANTAGES

1 *Precision Matching*. The intelligent routing mechanism ensures optimal alignment between question complexity and model capacity, improving accuracy for complex questions by over 20%.

2 *Dynamic Adaptability*. The parameter optimizer adjusts configurations in real time based on document characteristics, enabling stable cross-domain generalization without manual tuning.

3 *Closed-Loop Evolution.* The prompt engineering subsystem automates rule generation and application, reducing response latency to feedback to within minutes.

4 *Resource Efficiency.* Load-aware routing increases computational resource utilization by 15–25%, ensuring system responsiveness during peak loads.

2.6.4. PERFORMANCE METRICS

Table 5. Performance Metrics of Summarized Optimizer

METRIC TYPE	OBSERVED PERFORMANCE
ANSWER PRECISION	Accuracy on complex questions increased by 22%; Domain classification F1 score = 0.91
SYSTEM EFFICIENCY	Routing latency ≤ 100ms; Peak response time ≤ 500ms; Resource utilization improved by 20%
OPTIMIZATION GAINS	Parameter auto-tuning improved RAG retrieval NDCG@10 by 0.15; Prompt optimizer increased user satisfaction by 28%
ROBUSTNESS	Performance fluctuation ≤ 5% across 10+ industry domains; Parameter re-alignment latency ≤ 5 minutes upon document updates

3. **RAG Framework Comparison**

To address the complexities of cross-domain, multi-modal, and long-tail question answering scenarios, *5D-RAG* introduces an end-to-end optimization paradigm spanning five key stages: query rewriting, classification, tag-based retrieval, document retrieval, and answer generation. This unified design aims to improve both the performance and generalization capabilities of RAG systems across heterogeneous enterprise contexts. This experiment pursues two primary objectives:

1 To empirically validate the effectiveness of *5D-RAG* in terms of *retrieval quality* (measured by Hit@5 and MRR@5), answer accuracy (Fact-F1), and user satisfaction (CSAT);

2 To examine the marginal contributions and synergistic effects contributed by each optimizer module within the full *5D* pipeline.

3.1. Compared Systems

We selected three mainstream RAG frameworks as baselines, covering both classic and state-of-the-art approaches. The technical components and experimental positions of each system are summarized below:

Table 6. Comparison of RAG Frameworks

FRAMEWORK	TECHNICAL COMPOSITION	NOTES
NAIVE RAG	Fixed 512-token chunking + OpenAI text-embedding-ada + GPT-3.5	2023 baseline
ADVANCED RAG	BM25 + vector hybrid, HyDE、Re-ranker、Query2Doc	2024 best practice
MODULAR RAG	Level RAG + routing + multi-agent coordination	2025 SOTA
5D-RAG (OURS)	Rewriting → Classification → Tagging → Retrieval → Summarized Q&A Loop	Proposed in this report

3.2. Experimental Environment

3.2.1. BECHMARK

To evaluate the end-to-end performance of *5D-RAG* against mainstream frameworks, we constructed MIQA-25 (Multi-Industry QA 2025), a cross-domain benchmark dataset consisting of 150,000 industrial-grade QA pairs. It spans five verticals:

- Energy (30%)
- Telecommunications (25%)
- Office Automation (20%)
- Manufacturing (15%)
- Finance (10%)

The document structure exhibits the following characteristics:

- *Multi-modal complexity.* On average, each document includes 18% structured tables, 6% charts/figures, and 3% code blocks—requiring robust multi-modal understanding.
- *Long-context challenge.* Documents average 1,890 tokens, significantly longer than general-domain datasets (e.g., Natural Questions: ~512 tokens).

3.2.2. EVALUATION METRICS

We employed five quantitative dimensions to compare retrieval-augmented performance:

Table 7. Evaluation Metrics for RAG Comparison

METRIC TYPE	METRIC	DEFINITION
RETRIEVAL QUALITY	Top-5 Hit@k	Top-5 retrieval accuracy
	MRR@5	Mean Reciprocal Rank of the first relevant doc
ANSWER ACCURACY	Fact-F1	Entity-level F1 score for factual correctness
USER EXPERIENCE	CSAT	5-point user satisfaction rating
SYSTEM EFFICIENCY	RTFT	Response Time to First Token (ms)

3.2.3. EXPERIMENTAL ENVIRONMENT

All systems were tested under identical hardware and software conditions to ensure fairness:

- Inference Cluster: 8× NVIDIA RTX 4090 (NVLink)
- Framework Versions: vLLM 0.4.2 (optimized for continuous batching) + Milvus 2.4 (HNSW vector indexing)
- Re-ranking Model: BGE-Reranker-Large (1.3B parameters, cross-encoder architecture)

3.3. Results

In a cross-domain 5-shot setting, *5D-RAG* consistently outperformed all baselines across all key metrics, while maintaining manageable latency.

Table 8. Experimental Results Comparison of RAG Systems

SYSTEM→ METRIC ↓	NAIVE	ADVANCED	MODULAR	5D-RAG	Δ(5D VS. MODULAR)
HIT@5	61.4 %	78.1 %	84.6 %	90.7 %	+6.1 pt
MRR@5	0.52	0.68	0.74	0.81	+7 pt
FACT-F1	65.3 %	79.4 %	85.2 %	92.5 %	+7.3 pt
CSAT(USER)	3.3	3.9	4.2	4.6	+0.4
RTFT(ms)	1 920	1 450	1 380	1 360	-20 ms

Key Findings:

1 *Semantic Matching 5D-RAG*'s dynamic classification and multi-vector retrieval improved Hit@5 by 6.1 pts across industries.

2 *Modal Robustness* Tag filtering and Q&A joint optimization reduced multi-modal parsing errors (Fact-F1 +7.3 pts).

3 *Efficiency Balance* Full pipeline coordination introduced minimal latency (−20 ms), within statistical noise bounds.

3.4. Industry-Specific Ablation Studies

To evaluate the component-level impact of *5D-RAG* on cross-domain Q&A, we performed ablation studies on the top three industries from MIQA-25: Energy, Telecom, and Office Automation. Using Modular RAG as the base, we incrementally introduced each optimization module and compared final performance to the complete 5D-RAG.

Table 9. Fact-F1 by Industry (Ablation Study)

INDUSTRY	MODULAR	+REWRITE	+CLASSIFY	+TR^2	+RETRIEVE	+SUMMARIZED	5D-RAG
ENERGY	83.1%	84.6%	87.3%	89.5%	91.2%	93.0%	93.0%
TELECOM	85.4%	86.9%	88.8%	90.1%	91.9%	93.8%	93.8 %
OFFICE	86.7%	87.9%	89.1%	91.4%	92.7%	94.5%	94.5%

Key Findings:

1 *Generalization* The full *5D-RAG* achieved or exceeded the best ablation scores in all industries, confirming strong synergistic effects from end-to-end coordination.

2 *Component Necessity* The diminishing marginal returns (e.g., +QA adds only +1.9% in telecom) suggest the system is approaching Pareto-optimal design. Component-level impact varies by domain,

offering guidance for industry-specific customization (e.g., Energy benefits most from tagging; Office from answer optimization).

3.5. Key Module Contributions

Component-level analysis revealed the following contributions relative to the Modular RAG baseline:

1 *Rewriting Optimizer:*

- Reduced fuzzy/natural language queries from 28% to 9%, minimizing semantic noise.
- Cosine similarity to canonical intents improved from 0.11 → 0.87, boosting clarity.

2 *Classification Optimizer:*

- Reduced cross-domain noise recall rate by 42% (from 23% to 13%).
- Domain-specific document recall (e.g., Energy) improved by +6.7 pts.

3 *TR² Optimizer:*

- Refined document slicing granularity from 512-token to 128-token chunks.
- Answer token redundancy reduced by 34%, enhancing precision.

4 *Retrieval Optimizer:*

- Key passage hit rate in long documents increased by +9.4 pts.
- Table/code block retrieval rate improved by 13 pts, validating multi-modal capabilities.

5 *Summarized Optimizer* (Closed-loop):

- 12K feedback samples processed weekly via online scoring.
- CSAT increased by +0.22 within 2 weeks.
- Dynamic sample enrichment improved rewrite accuracy by +3.1 pts—forming a *positive feedback loop*.

3.6. Summary of Technical Advantages

1 *Performance Leadership* In full MIQA-25 scenarios, 5D-RAG outperformed Modular RAG by +7.3 pts in Fact-F1, +6.1 pts in Hit@5, with latency increase <2%.

2 *Domain Adaptability* Industry-level testing showed significant gains in highly structured domains (Energy, Telecom), validating the classification and tagging design.

3 *Scalability* The 5-layer loop (Rewrite–Classify–Tag Refine Retrieval –Retrieve–Answer) enables cross-domain, multi-modal, long-tail adaptation, offering a robust and extensible solution for industrial deployments.

4. Technical Capabilities

4.1. Core Technical Capabilities

This section outlines *the Platform's* foundational capabilities, structured along three principal dimensions: (1) *Multi-source heterogeneous knowledge acquisition and construction*, (2) *Knowledge reasoning*, and (3) *Intelligent knowledge applications*. Each capability is further decomposed into essential sub-

components, implementation methodologies, and measurable evaluation criteria, thereby enabling comprehensive system-level assessment.

4.1.1. MULTI-SOURCE HETEROGENEOUS KNOWLEDGE ACQUISITION AND CONSTRUCTION

Definition: This capability pertains to the transformation of fragmented, heterogeneous data—encompassing structured, semi-structured, and unstructured formats—into high-quality, semantically aligned, and structurally organized knowledge artifacts. It serves as the foundational layer of an enterprise-scale knowledge platform, ensuring that downstream reasoning and application modules operate on consistent and reliable knowledge representations.

Table 10. Core Metrics – Knowledge Acquisition and Construction

SUB-CAPABILITY	TECHNICAL IMPLEMENTATION	EVALUATION METRICS	DESCRIPTION
MULTI-SOURCE INGESTION COMPATIBILITY	Supports API, message queue, file-sync ingestion Compatible with relational DBs, NoSQL, and object storage	Supported source types: 10–15 Max daily throughput: 10–50 TB	Measures compatibility with heterogeneous data sources and ingestion scalability
ENTITY DISAMBIGUATION AND ALIGNMENT	Named Entity Recognition (NER) via rules + NLP Cross-source similarity matching (e.g., Jaccard + Embeddings)	Disambiguation accuracy: 85–92% Alignment latency: <50–100 ms / 10K records	Resolves name ambiguity (e.g., synonyms/homonyms) for consistent entity representation
ONTOLOGY MODELING	Visualized ontology editor Auto schema generation based on pattern mining	Classification accuracy: 88–95% Ontology construction efficiency: 20–50 classes/person-day	Ensures domain-aligned knowledge taxonomy
KNOWLEDGE EXTRACTION	Joint extraction models (entity + relation) Multimodal extraction (text, image OCR, speech ASR)	Extraction F1-score: 75–85% Text-image alignment accuracy: 80–90%	Extracts structured knowledge from complex documents
KNOWLEDGE FUSION	Neural network–based entity fusion Conflict resolution via attribute voting	Auto-merging coverage: 70–85% Conflict resolution rate: 80–90%	Constructs unified and trustworthy knowledge graphs

4.1.2. KNOWLEDGE REASONING

Definition: Knowledge reasoning encompasses logical inference, graph-based analysis, and causal discovery over structured knowledge representations. Its goal is to extract latent patterns, derive implicit facts, and enhance semantic understanding beyond explicitly stated information.

Table 11. Core Metrics – Knowledge Reasoning Capability

SUB-CAPABILITY	TECHNICAL IMPLEMENTATION	EVALUATION METRICS	DESCRIPTION
RULE-BASED INFERENCE	Rule engines (e.g., Drools) Ontology reasoning via SWRL	Rule trigger accuracy: 90–95%	Derives new facts based on domain-specific logical rules
GRAPH RELATIONSHIP MINING	Community detection algorithms multi-hop path inference	Hidden relation detection rate: 65–75%	Identifies latent associations among entities
GRAPH INDEX OPTIMIZATION	Compressed graph indexing Distributed graph computation	Recall: 90–95% Latency (billion-scale queries): <200–500 ms	Enables efficient large-scale graph exploration
CAUSAL REASONING	Bayesian network modeling temporal event graph analysis	Root cause inference accuracy: 70–80%	Supports root-cause diagnosis in complex business contexts

4.1.3. INTELLIGENT KNOWLEDGE APPLICATIONS

Definition: This capability focuses on the deployment of structured knowledge and reasoning outcomes into real-world business contexts. It enables intelligent retrieval, decision support, personalized knowledge delivery, and autonomous task execution. These applications collectively form a closed-loop knowledge service system, bridging knowledge construction and practical value realization.

Table 12. Core Metrics – Knowledge Reasoning Capability

SUB-CAPABILITY	TECHNICAL IMPLEMENTATION	EVALUATION METRICS	DESCRIPTION
SEMANTIC SEARCH	Vector-based retrieval engine Multi-hop QA using Graph Neural Networks (GNN)	Document localization accuracy: 82–88% Response time: <150–300 ms	Enables precise information retrieval under natural language queries
DECISION SUPPORT	Causal graph reasoning Visual decision path generation	Recommendation adoption rate: 60–75% Answer factuality: 90–92%	Provides interpretable suggestions for human-AI collaborative decision-making
KNOWLEDGE RECOMMENDATION	User profiling with knowledge associations Event-driven knowledge push	Click-through rate (CTR): 15–25%	Achieves personalized knowledge delivery via user–knowledge matching
Automated Knowledge Services	RPA + knowledge API integration Low-code orchestration	Process efficiency improvement: 30–50%	Automates task execution based on knowledge triggers

4.2. Additional Capabilities

This chapter presents three extended capabilities of *the platform*: *full-lifecycle knowledge management*, *multi-agent orchestration*, and *enterprise-grade security adaptation*. These capabilities demonstrate the system's engineering scalability, governance feasibility, and deployment readiness in complex enterprise environments.

4.2.1. Full-Lifecycle Knowledge Management

The platform supports an end-to-end pipeline for knowledge asset creation, validation, application, and archiving. It incorporates automated governance and high-throughput frameworks to ensure quality, consistency, and adaptability across heterogeneous data sources.

1 *Automated Quality Control.* Implements rule-based and model-assisted verification of factuality, timeliness, and semantic coherence. A closed-loop governance loop corrects noisy or low-value content.

2 *Multimodal Data Compatibility.* Ingests unstructured (*e.g., .txt, .md, .doc/docx, .pdf, .ppt/.pptx, .mp4, .wav, .jpg, .png*) and structured data (*e.g., .csv, SQL)* through a unified parser, enabling heterogeneous source fusion.

3 *Metadata-Driven Explainability.* Captures source, version, usage statistics, and quality scores as structured metadata, enhancing knowledge transparency and reusability.

4 *Traceable Output Generation.* Aligns retrieved answers with original document fragments. Complex documents are auto-segmented and indexed to support verifiable outputs.

5 *High-Throughput Retrieval Engine.* Delivers sub-second query latency on a 100B-scale knowledge graph, supporting concurrent inference with horizontal and vertical scaling.

4.2.2. MULTI-AGENT ORCHESTRATION

The platform supports collaborative workflows across task, retrieval, and reasoning agents. A unified orchestration engine manages toolchains and shared knowledge resources for consistent and coordinated agent behavior.

1 *Composable APIs.* Encapsulates core capabilities (e.g., retrieval, semantic routing, graph traversal) into reusable APIs compatible with *LangChain* and *AutoGen*.

2 *Dynamic Knowledge Service Orchestration*. Automatically assembles toolchains based on tasks, supporting cross-module reasoning, validation, and conflict resolution. This transforms isolated knowledge services into scenario-based intelligence.

3 *Unified Knowledge Context.* Agents share a real-time knowledge base through a unified access point, ensuring that task-oriented, decision-making, and operational agents collaborate on a consistent factual basis—eliminating redundancy and version inconsistencies.

4 *Conflict Handling.* Applies version locks and arbitration rules to manage concurrent updates, preserving asset integrity and logical consistency.

5 *Feedback Loop Optimization.* Logs user interactions, decision outcomes, and tool usage. Low-quality knowledge or ineffective tools are automatically identified and refined.

6 *Memory and Privacy Management.* Maintains short- and long-term memory for personalization. In privacy mode, sensitive data is excluded from storage and retrieval.

4.2.3. ENTERPRISE SECURITY ADAPTATION

To ensure enterprise-grade trust and compliance, the system establishes an end-to-end protection framework covering identity, behavior, and transaction security.

1 *Unified Access Control.* Implements role- and attribute-based access control (RBAC + ABAC) for fine-grained, policy-driven authorization across layers..

2 *Dynamic Trust Scoring*. Continuously evaluates risk in real-time using behavioral features (e.g., access frequency, task criticality), enabling adaptive privilege management.

3 *Transactional Consistency.* Ensures atomicity and traceability of multi-agent updates via version verification and conflict-resolution protocols.

4 *Comprehensive Auditing.* Tracks every action—from API invocation to agent outputs—with button-level granularity. Real-time risk scoring enables threat detection and forensic analysis.

References

Jianlv, C., Shitao, X., Peitian, Z., Kun, L., Defu, L., & Zheng, L. (2024) *BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation*, Annual Meeting of the Association for Computational Linguistics: 2318-2335.

Jueyou, L., Chuanye, G., Zhiyou, W., & Tingwen, H. (2022) *Online Learning Algorithm for Distributed Convex Optimization with Time-Varying Coupled Constraints and Bandit Feedback*, IEEE Transactions on Cybernetics, 52.2: 1009-1020.

Jianmo, N., Gustavo Hernandez, A., Noah, C., Ji, M., Keith B., H., Daniel, C., & Yinfei, Y. (2022) *Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models*, Computing Research Repository: 1864-1874.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Suleyman, M. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459-9474.

Patrick, L., Ethan, P., Aleksandra, P., Fabio, P., Vladimir, K., Naman, G., Heinrich, K., Mike, L., Wen-tau, Y., Tim, R., Sebastian, R., & Douwe, K. (2020) *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.*, Conference on Neural Information Processing Systems, 33: 9459-9474.

Soyeong, J., Jinheon, B., Sukmin, C., Sung Ju, H., & Jong C., P. (2024) *Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models Through Question Complexity*, Computing Research Repository: 7036-7050.

Shitao, X., Zheng, L., Peitian, Z., Niklas, M., Defu, L., & Jian-Yun, N. (2024) *C-Pack: Packed Resources for General Chinese Embeddings*, SIGIR 2024.

Woosuk, K., Zhuohan, L., Siyuan, Z., Ying, S., Lianmin, Z., Cody Hao, Y., Joseph E., G., Hao, Z., & Ion, S. (2023) *Efficient Memory Management for Large Language Model Serving with PagedAttention*, PROCEEDINGS OF THE TWENTY-NINTH ACM SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES, SOSP 2023: 611-626.

Wenqi, F., Yujuan, D., Liangbo, N., Shijie, W., Hengyun, L., Dawei, Y., Tat-Seng, C., & Qing, L. (2024) *A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models*, Computing Research Repository: 6491-6501.

Yunfan, G., Yun, X., Meng, W., & Haofen, W. (2024) *Modular RAG: Transforming RAG Systems into LEGO-like Reconfigurable Frameworks*, Computing Research Repository, abs/2407.21059

Yunfan, G., Yun, X., Xinyu, G., Kangxiang, J., Jinliu, P., Yuxi, B., Yi, D., Jiawei, S., Meng, W., & Haofen, W. (2023) *Retrieval-Augmented Generation for Large Language Models: A Survey*, Computing Research Repository, abs/2312.10997

Yu A., M., & D. A., Y. (2018) *Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 42.4: 824-836.

Zhuocheng, Z., Yang, F., & Min, Z. (2025) *LevelRAG: Enhancing Retrieval-Augmented Generation with Multi-hop Logic Planning over Rewriting Augmented Searchers*., Computing Research Repository, abs/2502.18139