Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Hierarchical Semantic Structuring and Evolutionary Query Optimization Framework for Domain-Specific Retrieval- Augmented Generation

Komal Paul
Adamas University
Tamal Ghosh
dr.tamal.ghosh@outlook.com
Adamas University
Sajal Saha
Adamas University

Article

Keywords: Large Language Models, Domain-Specific Intelligence, Structured Prompt Engineering, Prompt Optimization, OSMOSIS Framework, Camouflage Optimization Algorithm

Posted Date: November 23rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11137327/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Hierarchical Semantic Structuring and Evolutionary Query Optimization Framework for Domain-Specific Retrieval-Augmented Generation

Komal Paul[1], Tamal Ghosh[2]*, Sajal Saha[3]
[1, 2, 3] Department of Computer Science and Engineering, Adamas University
AKC Campus, Barasat- Barrackpore Road, Jagannathpur, PIN - 700126, India
Email: dr.tamal.ghosh@outlook.com, Ph: +91-9163253061
*Corresponding Author

Abstract

Large language models (LLMs) have demonstrated strong capabilities across natural language processing tasks; however, deploying them in domain-specific and edge-constrained environments remains challenging because of semantic ambiguity, retrieval noise, computational overhead, and inadequate contextual grounding. Existing retrieval-augmented generation (RAG) frameworks often rely on static retrieval mechanisms and unstructured prompts, while current prompt engineering approaches generally lack adaptive optimization and edge compatibility. To address these limitations, this study proposes a multi-stage prompt optimization framework integrating the OSMOSIS semantic structuring engine with the Camouflage Optimization Algorithm (COA). OSMOSIS transforms ambiguous natural-language prompts into schema-constrained JSON representations, while COA adaptively refines retrieval relevance through evolutionary optimization. Experimental evaluation across legal, medical, financial, and career counseling domains demonstrates superior performance over seven baselines, including DSPy, Self-RAG, HyDE, and GraphRAG, achieving a BERTScore F1 of 0.813, ROUGE-L of 0.717, and 95.4% task accuracy. Statistical validation confirms the framework's robustness and generalizability.

Keywords: Large Language Models; Domain-Specific Intelligence; Structured Prompt Engineering; Prompt Optimization; OSMOSIS Framework; Camouflage Optimization Algorithm

1. Introduction

The emergence of large language models (LLMs) has been widely regarded as a transformative development in artificial intelligence (AI), owing to the high-level natural language understanding, contextual reasoning, and generative capabilities that such models have been found to exhibit [1–3]. Transformer-based architectures, including GPT, BERT, LLaMA, and Gemma, have been extensively applied across tasks such as text summarization, conversational AI, information retrieval, and decision-making. As LLM-based applications have grown rapidly in scope and scale, considerable attention has been directed toward the development of intelligent language systems tailored to domain-specific contexts, including healthcare, finance, legal analysis, cybersecurity, and career guidance [4–7].

Despite the superior capabilities demonstrated by general-purpose LLMs, significant challenges have been identified when such models are employed in domain-specific settings [1]. Foundational models are predominantly trained on unstructured, internet-scale datasets, from which domain-specific expertise is largely absent [8]. Consequently, outputs generated by these models have been observed to be ambiguous, hallucinated, poorly grounded, and semantically inconsistent. Such shortcomings are considered particularly critical in high-stakes domains such as medical diagnosis and legal analysis, where factual accuracy and traceability are regarded as essential requirements [9–11].

Concurrently, the growing demand for Edge AI deployment has introduced an additional layer of constraints. Edge-based systems are characterized by limited computational capacity, restricted memory availability, constrained power consumption, and stringent latency requirements [12, 13]. Although

cloud-based language intelligence infrastructures are recognized for their strong computational capabilities, they have been associated with elevated security risks arising from network dependency and increased response latency [1].

Retrieval-Augmented Generation (RAG) architectures have been proposed as an effective means of enhancing contextual awareness by incorporating external knowledge retrieval into the LLM inference pipeline [9, 14]. However, conventional RAG frameworks rely heavily on unstructured user prompts and static retrieval embeddings, resulting in retrieval noise and inadequate semantic alignment between retrieved documents and the posed query. Recent efforts in prompt engineering have attempted to mitigate these issues through structured prompting, query reformulation, and retrieval refinement; yet, these approaches have been shown to fall short of providing comprehensive solutions [15–17].

Several critical limitations have been identified in existing LLM frameworks designed for domain-specific deployment. First, raw user prompts have been observed to carry inherent semantic ambiguities [6]. Second, static frameworks have been found incapable of dynamically optimizing query representations based on contextually relevant signals [18]. Third, cloud-dependent LLM platforms have been noted to introduce privacy vulnerabilities and computational burdens that cannot be accommodated by resource-constrained edge devices [1, 12].

Several research gaps have been identified through a critical review of the existing literature [9, 19]. First, most conventional RAG systems perform retrieval directly on unstructured user prompts without decomposition into semantically meaningful components, often limiting contextual precision and semantic alignment during retrieval [20, 21]. Although structured prompting approaches have demonstrated improvements in prompt organization and reasoning consistency, adaptive retrieval optimization mechanisms remain insufficiently explored in such frameworks [22, 23]. Similarly, optimization-based retrieval strategies have shown potential for improving retrieval relevance; however, structured semantic decomposition has rarely been incorporated into these approaches in a unified manner [15, 18].

Furthermore, recent retrieval and prompt optimization studies have not been comprehensively benchmarked against advanced retrieval-aware systems such as DSPy, Self-RAG, GraphRAG, and HyDE, particularly under edge-constrained deployment environments where computational efficiency becomes a critical requirement [24–28]. In addition, comprehensive evaluation protocols that integrate ablation analysis, statistical validation, and multi-dimensional performance assessment—including latency, memory consumption, semantic quality, contextual consistency, and retrieval precision—remain comparatively limited in existing domain-specific retrieval studies [13, 29]. These limitations collectively motivate the development of a structured and adaptive retrieval framework that is capable of improving semantic grounding while maintaining practical feasibility for resource-constrained deployment.

In light of the aforementioned limitations, the following contributions are presented in this study:

- A dual-stage prompt refinement framework is proposed, integrating the OSMOSIS structuring component with the Camouflage Optimization Algorithm (COA).
- A schema-constrained semantic decomposition mechanism is introduced for the transformation of ambiguous user prompts into structured JSON representations.
- An adaptive optimization system is developed, through which query weights and semantic retrieval relevance are dynamically adjusted based on user feedback.
- A lightweight vector memory-based retrieval pipeline is constructed, incorporating quantized inference to support deployment on resource-constrained edge devices.
- Evaluations are conducted across legal, medical, financial, and career counselling datasets.
- Extensive benchmarking is performed against established baselines, including DSPy, Self-RAG, HyDE, GraphRAG, query rewriting, and reranking approaches.
- Pareto trade-off analysis, heatmap benchmarking, statistical validation, and quantitative ablation studies are presented to validate the effectiveness of the proposed framework.

The remainder of this paper is organized as follows. Section 2 presents a review of related work, encompassing prior studies on RAG architectures, prompt engineering, edge AI deployment, and domain-specific LLM adaptation. Section 3 describes the proposed methodology, detailing the OSMOSIS structuring module, the Camouflage Optimization Algorithm (COA), and the lightweight

retrieval pipeline. Section 4 outlines the experimental setup, results of comparative experiments, ablation studies, Pareto trade-off assessments, and heatmap benchmarks. Finally, Section 5 concludes the paper by summarizing the key contributions and their broader implications.

2. Related Work

2.1 Domain-Specific Large Language Models

Domain-specific LLMs have gained prominence for improving contextual reasoning and semantic accuracy in specialized application domains. Representative models such as Med-PaLM, BloombergGPT, and Legal-BERT demonstrate that domain alignment can improve factual consistency, contextual grounding, and task-specific performance in healthcare, finance, and legal environments [1, 11, 30–32]. In particular, medical AI models have shown potential for supporting clinical reasoning and medical knowledge retrieval. Financial and legal models exhibit improved capabilities in financial analysis and jurisdiction-specific language understanding.

Despite these advantages, domain-specific adaptation strategies frequently require substantial computational resources, large-scale domain corpora, and costly fine-tuning mechanisms, thereby limiting practical deployment in resource-constrained environments. Furthermore, many domain-specific systems still rely on conventional retrieval mechanisms that lack adaptive optimization and structured semantic decomposition, limiting contextual flexibility and retrieval robustness in complex domain-specific reasoning tasks.

2.2 Retrieval-Augmented Generation

RAG integrates external knowledge retrieval with LLM inference to improve factual grounding and reduce hallucination during response generation [27, 33]. In a conventional or vanilla RAG pipeline, semantically relevant documents are retrieved through embedding-based similarity matching and subsequently incorporated into the generation process to provide contextual evidence. Traditional RAG systems are factually inconsistent and vulnerable to retrieval noise, semantic mismatch, and context dilution, which results from inadequate query representations and static retrieval embeddings [9, 34].

To address these limitations, several advanced retrieval-aware frameworks have recently been proposed, including Self-RAG, GraphRAG, HyDE, and DSPy [24–27]. Self-RAG introduces retrieval-aware self-reflection to iteratively improve contextual grounding during generation, whereas GraphRAG incorporates graph-structured knowledge representations to strengthen contextual connectivity and relationship-aware retrieval. In contrast, HyDE improves retrieval robustness using hypothetical document embedding generation that better aligns retrieval candidates with user intent [27]. Despite these advances, most retrieval-aware approaches still emphasize retrieval enhancement without integrating structured semantic decomposition and adaptive optimization within a unified, edge-compatible framework.

2.3 Structured Prompt Engineering

Structured prompting techniques have been increasingly explored to improve semantic organization and reasoning consistency by transforming natural language instructions into constrained and interpretable representations [6, 23]. Such approaches improve contextual clarity and reduce ambiguity through explicit prompt organization and schema-aware reasoning structures. In particular, modular prompting frameworks such as DSPy employ declarative decomposition strategies to improve reasoning quality and task orchestration across complex inference pipelines. Similarly, schema-driven and JSON-based prompting methods enhance prompt interpretability and semantic consistency by enforcing structured representations during retrieval and generation.

Despite these advantages, most existing structured prompting approaches focus primarily on prompt decomposition and do not incorporate adaptive retrieval optimization mechanisms. Consequently, retrieval effectiveness may remain sensitive to contextual ambiguity and static embedding behavior,

resulting in suboptimal document retrieval and insufficient semantic alignment in domain-specific reasoning scenarios[9].

2.4 Prompt Optimization Techniques

Prompt optimization techniques have been increasingly explored to improve large language model (LLM) performance through adaptive refinement of prompt structures using heuristic strategies, evolutionary algorithms, and reinforcement learning approaches [6, 15, 18]. In parallel, rewriting of query mechanisms can reformulate user prompts to improve retrieval relevance, contextual grounding, and semantic alignment during information retrieval and generation tasks. Such approaches have demonstrated the potential to improve retrieval quality and response consistency by refining semantic query representations.

The COA is a nature-inspired optimization method designed to balance exploration and exploitation through adaptive search behavior under dynamic retrieval environments [35]. However, current optimization methods can enhance search capability only and do not consider semantic decomposition and context-aware retrieval mechanisms.

2.5 Research Gap Summary

A critical review of the literature indicates that existing retrieval-enhanced LLM frameworks remain fragmented in semantic structuring, adaptive optimization, contextual retrieval, and deployment feasibility. Conventional RAG systems primarily emphasize contextual retrieval but frequently operate on unstructured prompts without semantic decomposition. Structured prompting approaches improve interpretability and reasoning consistency; however, adaptive retrieval optimization is often insufficiently integrated. Similarly, optimization-driven retrieval methods improve query relevance but rarely incorporate schema-constrained semantic representations within a unified framework.

Furthermore, recent retrieval-aware systems such as DSPy, Self-RAG, GraphRAG, and HyDE demonstrate strong capabilities under specific retrieval settings, yet comprehensive evaluation across semantic quality, computational efficiency, contextual grounding, and edge compatibility remains comparatively limited (Table 1). These observations collectively motivate a unified framework that integrates structured semantic decomposition, adaptive optimization, contextual retrieval, and lightweight deployment for domain-specific inference. Accordingly, the proposed OSMOSIS–COA framework addresses these limitations through a multi-stage retrieval and optimization pipeline.

Table 1. Comparative Analysis of Existing Retrieval Frameworks

Method	Structured Prompting	Optimization	Context Retrieval	Edge Compatible
Vanilla RAG	✗	✗	✓	△
DSPy	✓	△	✓	△
Self-RAG	✗	✓	✓	△
GraphRAG	✗	✗	✓✓	✗
HyDE	✗	✗	△	△
Proposed OSMOSIS–COA	✓	✓	✓	✓

Legend: ✓ = supported; △ = partial/limited; ✗ = unsupported; ✓✓ = strong capability

3. Proposed Framework

3.1 Overall Architecture

The proposed framework is a modular, multi-stage architecture for semantic retrieval and domain-specific LLM inference in edge-compatible environments (Figure 1). The architecture comprises five main components:

- Domain-aware query routing
- OSMOSIS structuring engine
- Context-based retrieval layer
- COA optimization engine
- Inference pipeline

The workflow begins with user-provided raw prompts. The queries are classified using semantic approaches and directed through different retrieval pipelines for different domains. The OSMOSIS component translates unstructured prompts into structured JSON objects based on the schema. The system stores semantic contexts as vectors in memory for retrieval. Next, the COA component optimizes the retrieval weights and semantic relevance.

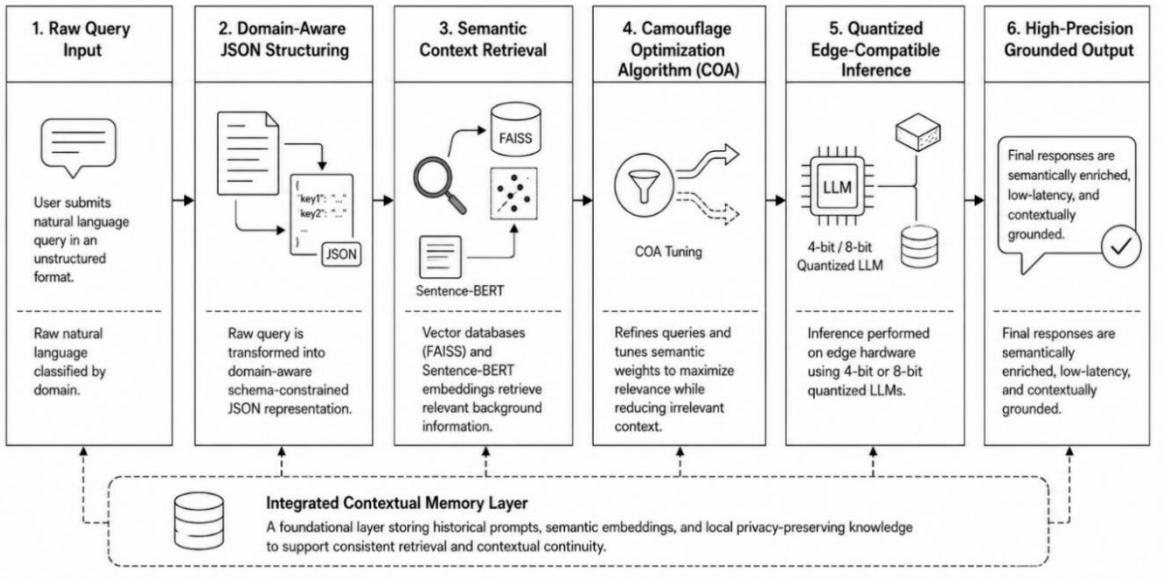

Figure 1. Overall Framework Architecture

3.2 OSMOSIS Structuring Engine

The OSMOSIS engine is the system's semantic intelligence layer (Figure 2). The main goal of the OSMOSIS engine is to turn noisy, ambiguous raw inputs into structured inputs understandable by machines [36, 37].

This can be expressed mathematically as follows:

$$Q_s = F(Q_r)$$

Where:

- Q_r is the raw input query
- Q_s is the structured input query
- F stands for the semantic structuring function

The structured schema consists of the following:

- Intent
- Key Topics
- Expected Output Type
- Contextual Constraints

3.2.1 Standardized Schema Modeling

The framework decomposes natural language prompts into structured semantic dimensions. This decomposition ensures consistent interpretation throughout retrieval.

3.2.2 Schema-Constrained Decoding

This framework uses lightweight decoding techniques to ensure JSON outputs remain valid. The technique helps avoid irrelevant conversation generation while improving retrieval accuracy.

3.2.3 Linguistic Noise Reduction

The preprocessing module eliminates unnecessary fillers and non-informational words. This process improves embedding density and semantic coherence.

3.2.4 Pattern-Based Intent Detection

Rule-based heuristics will classify procedural, analytical, comparative, and descriptive prompts. This classification helps ensure contextual routing and retrieval.

3.2.5 Hardware-Efficient Deployment

The OSMOSIS engine uses 4-bit and 8-bit quantized inference techniques to target resource-constrained edge devices.

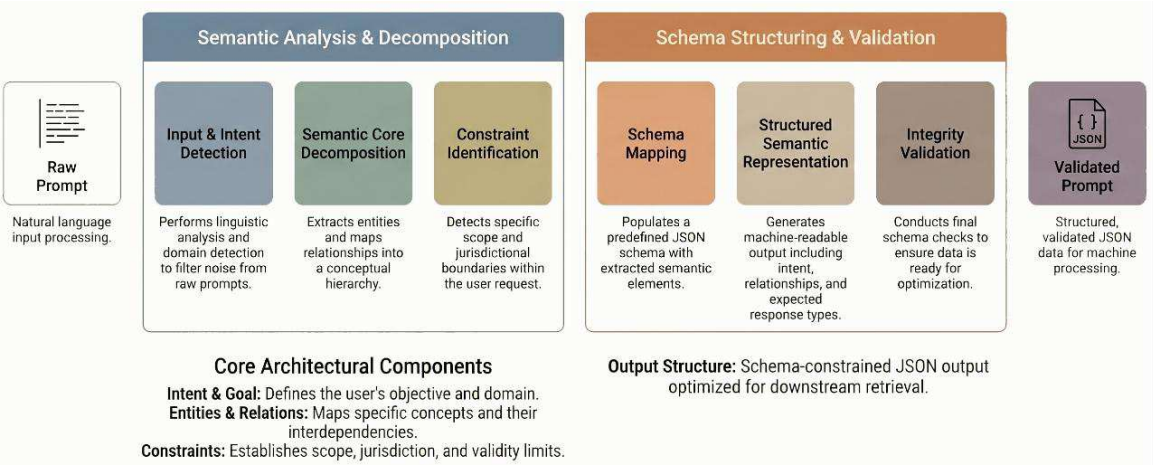

Figure 2. OSMOSIS Semantic Structuring Workflow

3.3 Camouflage Optimization Algorithm (COA)

The COA component provides the optimization intelligence in the workflow shown in Figure 3. While OSMOSIS focuses on semantic structuring, COA optimizes the search relevance. The COA algorithm can be studied in a recent article by [35, 38].

3.3.1 Query Expansion

The optimization engine creates several potential variations of the query through:

- keyword decomposition,
- semantic expansion,
- bigram extraction, and
- contextual enrichment.

3.3.2 Adaptive Weight Initialization

The initial weights for candidate retrievals are generated with Latin Hypercube Sampling (LHS).

3.3.3 Fitness Function

The fitness function for optimization assesses the relevance of retrieval based on cosine similarity:

$$Fitness = argmax(\sum Cosine(Q_e, D_r) - \lambda C)$$

where:

- Q_e refers to the enriched query embeddings
- D_r refers to the retrieved document embeddings

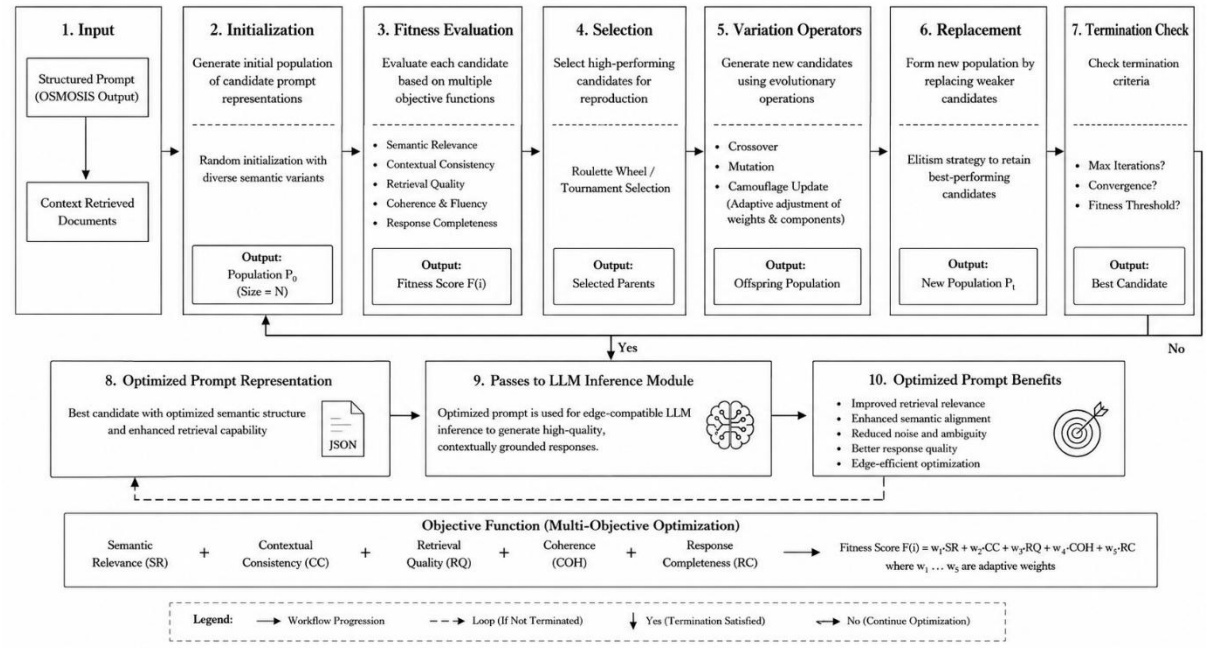

Figure 3. COA Optimization Workflow

3.3.4 Camouflage Density Mechanism

The camouflage mechanism ensures an optimal balance between diversity and relevance.

3.3.5 Escape Strategy

The escape strategy is employed when the optimization process hits a roadblock by randomly redistributing weights.

3.3.6 Predator Local Search

Optimization is further enhanced through local neighborhood refinement of candidate queries.

3.3.7 COA-Based Prompt Refinement

Algorithm 1: Camouflage Optimization-Based Query Refinement for OSMOSIS-Structured Retrieval

Require: Structured prompt $S_{structured}$, candidate expansion set E, vector database VDB
Ensure: Optimized query representation $Q_{optimized}$
1: **Generate** semantic candidate expansion set:
$E = \{e_1, e_2, e_3, ..., e_n\}$
2: **Initialize** adaptive weight vectors using Latin Hypercube Sampling (LHS):
$W = \{w_1, w_2, w_3, ..., w_n\}$
3: **for** t = *1* to T_{max} **do**
4: Compute weighted semantic query representation:
$Q_t = \sum_{i=1}^{n} w_i \cdot \text{Embedding}(e_i)$
5: Retrieve top-k contextual documents:
$D_t = \text{Retrieve}(Q_t, VDB)$
6: Evaluate retrieval fitness:

$F_t = -\text{MeanCosineSimilarity}\ (Q_t, D_t)$
7: Apply camouflage density adjustment to adapt semantic weights
8: **if** fitness stagnation is detected **then**
9: Perform escape operation for exploration enhancement
10: **end if**
11: Apply predator local search around the current best candidate
12: Update optimal weight vector:
$W^* \leftarrow \arg\ min_W (Ft)$ ▷ Store weights yielding lowest fitness score
13: **end for**
14: Generate optimized query representation:
$Q_{optimized} \leftarrow \sum_{i=1}^{n} w_i^* \cdot \text{Embedding}(e_i)$
15: **return** $Q_{optimized}$

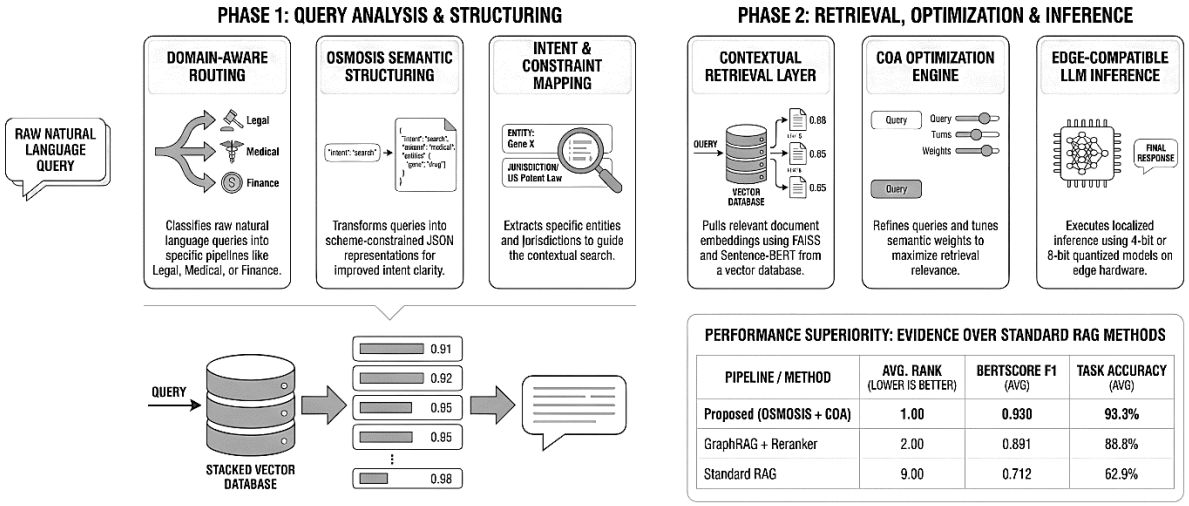

Figure 4. End-to-End Multi-Stage Workflow

3.4 Contextual Retrieval Layer

A vector database has been incorporated to improve semantic memory retrieval. The OSMOSIS model's structured outputs are embedded and saved to localized vector memory.

The FAISS vector index will be used primarily for this purpose. Sentence-BERT embeddings are used for semantic similarity searches. Contextual memory improves retrieval continuity and domain understanding.

3.5 Edge-Friendly Deployment Approach

An approach is adopted to deploy the model at the edge for localized inference, including quantized LLMs, lightweight vector search, localized semantic memory, and minimal reliance on the cloud.

3.6 Multi-Stage Prompt Structuring and Optimization Pipeline

Algorithm 2 Multi-Stage Prompt Structuring and Optimization Pipeline

Require: Raw user query Q_{raw}, vector database VDB, domain-specific language models LLM_{domain}
Ensure: Optimized response R_{opt}

1: Receive raw query Q_{raw}
2: $Q_{clean} \leftarrow \text{RemoveNoise}\ (Q_{raw})$
3: $D_{target} \leftarrow \text{DomainRouter}\ (Q_{clean})$

4: Forward query to corresponding domain-specific LLM (LLM_{target})
5: $S_{structured} \leftarrow$ OSMOSIS(Q_{clean}, LLM_{target})
6: Convert structured representation Sstructured into JSON schema J_{schema}
7: $Docs_{init} \leftarrow$ VectorRetrieve (J_{schema}, VDB)
8: $E \leftarrow$ QueryExpansion ($Docs_{init}$, J_{schema})
9: $W \leftarrow$ InitializeWeights (E)
10: $S_{score} \leftarrow$ CosineSimilarity (E, VDB)
11: $Q_{opt} \leftarrow$ COAOptimizer(W, E, $Sscore$) ▷ Applies Camouflage Optimization
12: $Docs_{opt} \leftarrow$ Retrieve(Q_{opt}, VDB)
13: $C_{ctx} \leftarrow$ ContextAggregation(Q_{opt}, $Docs_{opt}$)
14: $R_{opt} \leftarrow$ GenerateResponse(C_{ctx}, LLM_{target})
15: Store structured outputs in vector memory VDB
16: **return** R_{opt}

3.6.1 Prompt Template Standardization

To ensure experimental fairness, all retrieval pipelines used a standardized prompt formulation strategy. Only retrieval methodology and optimization behavior varied among baselines, while prompt intent remained consistent.

The generalized prompt structure employed across all experiments is given below:

User Query:
{Raw User Input}

Retrieved Context:
{Top-k Retrieved Documents}

Instruction:
Generate a contextually grounded response that maintains semantic consistency and minimizes hallucination.

For the OSMOSIS framework, the raw prompt was first converted into a schema-constrained JSON representation consisting of:

- intent,
- key topics,
- contextual constraints,
- expected response type, and
- semantic priority.

The structured prompt was subsequently optimized using the COA refinement stage before contextual retrieval. This standardization prevented performance inflation from prompt-engineering bias and ensured a fair comparative evaluation.

3.7 Retrieval Optimization Formulation

The retrieval relevance is then improved by proposing a composite retrieval score:

$$S_{total} = \alpha S_{semantic} + \beta S_{context} + \gamma S_{retrieval}$$

where:

- $S_{semantic}$ represents semantic similarity,
- $S_{context}$ denotes contextual alignment,
- $S_{retrieval}$ indicates retrieval confidence,
- and α, β, γ are adaptive weighting coefficients.

COA dynamically tunes coefficient values to maximize semantic relevance while minimizing retrieval noise. The optimization function is stated by the formula below:

$$Q^* = \arg\max_{Q_i}(\sum_{i=1}^{n} Sim(Q_i, D_i) - \lambda C_i)$$

where Q_i represents candidate query embeddings, $D_i$denotes retrieved document embeddings, represents computational cost, and λ is the regularization coefficient that controls optimization overhead. The proposed mathematical model balances semantic relevance, contextual relevance, and computational performance. With structural semantics and an optimization approach, the system enhances its retrieval precision and reasoning capabilities. Figure 4 illustrates the end-to-end multi-stage workflow.

3.8 Novelty and Research Significance

Unlike standard RAG systems, which use raw input prompts for retrieval, the proposed architecture presents a unified multi-stage framework with semantic decomposition, optimization, contextual retrieval, and edge-compatibility capabilities built into a single domain-specific LLM workflow. By translating ambiguous natural language prompts into constrained structured representations with the OSMOSIS engine, the proposed framework achieves much better contextual alignment, retrieval accuracy, and interpretability by reducing the ambiguity of embeddings and retrieved documents.

A key novelty of the proposed work is combining COA with semantic structuring. Many prompt-optimization techniques address either retrieval improvement or input reformulation. The proposed framework provides for complementary stages of semantic structuring and adaptive optimization. While the OSMOSIS module improves semantic structuring, the COA engine uses evolutionary search to adjust embeddings to maximize retrieval relevance and adaptively optimize weights.

A key advantage of the presented architecture is edge compatibility, enabled by the efficient design of its components. Unlike most retrieval-enhanced LLMs, which rely on costly cloud-based infrastructure, the proposed framework uses lightweight retrieval, model quantization, and local contextual knowledge to enable efficient edge deployment.

Furthermore, the proposed framework undergoes statistical robustness validation through Friedman rank testing, Wilcoxon signed-rank analysis, and critical difference benchmarking to ensure performance gains are statistically reliable rather than stochastic.

4. Results, Analysis and Discussion

4.1 Experimental Settings

The experiments were performed on four datasets (Table 2) from specific domains.

i) The customized legal dataset of India was created based on the Constitution of India papers, along with legal Q&A pairs [39].

ii) Medical question-answering datasets based on symptom description, diagnosis, and clinical reasoning problems were employed [40].

iii) Databases for financial reasoning problems, including banking, investments, and budgeting questions, were tested [41].

iv) Databases related to career counseling, comprising education-related [42].

Table 2. Dataset Summary

Dataset	Source	Format	Domain Type	Purpose
Indian Legal Dataset	Constitution of India + Custom PDF Pipeline	PDF → JSON	Legal	Structured legal reasoning
Medical Dataset	Online Repository	QA + Clinical Text	Healthcare	Semantic medical retrieval
Finance Dataset	Online Repository	Reports + Queries	Finance	Financial reasoning
Career Counseling Dataset	Online Repository	Dialogues + QA	Career Guidance	Personalized recommendation

Table 3 summarizes the hardware and software environment used to implement and evaluate the proposed framework. The experiments used an edge-compatible computational setup to ensure practical deployment feasibility under resource-constrained conditions. This configuration enabled efficient evaluation of semantic retrieval quality, optimization latency, and memory utilization across multiple benchmark pipelines.

Table 3. Hardware Configuration

Component	Configuration
CPU	Intel Core i7 / Ryzen 7
RAM	8-16 GB
GPU	Optional NVIDIA RTX
Quantization	4-bit / 8-bit
Vector Database	FAISS
Framework	Python + Transformers
Embedding Model	Sentence-BERT
Deployment	Edge-Compatible Local Inference

To ensure reproducibility and evaluate the scalability of the proposed OSMOSIS–COA framework under varying retrieval conditions, a detailed parametric configuration analysis was performed (Table 4). The experiments were conducted by maintaining identical embedding models, retrieval strategies, chunk sizes, and inference settings across all benchmark pipelines.

The vector database size was progressively increased to analyze retrieval scalability under edge-compatible conditions (Table 5). Experiments were conducted using FAISS indexing and Sentence-BERT embeddings while monitoring inference latency, memory utilization, and retrieval accuracy.

This experiment aims to determine whether the proposed structured prompt optimization pipeline can maintain semantic consistency and computational efficiency as contextual memory requirements increase.

Table 4. Experimental Parameter Settings

Component	Parameter	Value
Embedding Model	Sentence-BERT	all-MiniLM-L6-v2
Vector Database	FAISS Index	IndexFlatIP
Chunk Size	Text Segment Length	512 tokens
Chunk Overlap	Overlap Size	64 tokens
Top Retrieval	Top-k	5
Similarity Metric	Retrieval Score	Cosine Similarity
Quantization	LLM Compression	4-bit / 8-bit
COA Population Size	N	30
COA Iterations	T	50
Escape Threshold	Stagnation Epoch	10
Local Search Radius	Predator Search	0.1
Optimization Objective	Fitness	Semantic similarity maximization

Hardware	CPU	Intel i7 / Ryzen 7
RAM	Memory	16 GB
Framework	Environment	Python 3.11 + Transformers
Statistical Runs	Repetitions	5

Table 5. Vector Database Scalability Analysis

Vector DB Size	Retrieval Time (s)	Memory Usage (MB)	Task Accuracy (%)	BERTScore F1
10K	1.21	356	94.6	0.782
50K	1.48	402	93.9	0.775
100K	1.86	471	92.8	0.768

The scalability evaluation demonstrates that the proposed OSMOSIS–COA framework maintains stable semantic retrieval quality as vector database sizes increase (Table 5). Although retrieval latency and memory utilization increase gradually with larger vector memory, semantic accuracy remains comparatively stable.

The findings indicate that the proposed retrieval mechanism exhibits robust contextual retrieval performance under edge-compatible deployment conditions, thereby validating its suitability for scalable domain-specific inference environments.

The proposed framework was evaluated against several state-of-the-art retrieval and prompt optimization methods to ensure comprehensive performance benchmarking. These baselines include conventional retrieval systems, retrieval-augmented generation frameworks, structured prompting approaches, and optimization-driven pipelines such as DSPy, HyDE, Self-RAG, GraphRAG, and Prompt Optimization. The comparison enables assessment of the proposed method in terms of semantic quality, contextual relevance, computational efficiency, and robustness across diverse domain-specific tasks.

The fairness and reproducibility of comparative benchmarking were ensured through standardized implementation settings across all retrieval pipelines. Because retrieval-based systems are highly sensitive to embedding strategy, chunking, prompt construction, and retrieval depth, we maintained identical foundational configurations whenever possible to reduce experimental bias. All retrieval pipelines were implemented under the same localized edge-compatible environment using identical embedding models, retrieval chunk size, and vector indexing configuration. Sentence-BERT (all-MiniLM-L6-v2) was employed as the shared semantic embedding model across all methods to maintain embedding consistency. The FAISS vector database was used uniformly for retrieval indexing, while all experiments were executed using Python and Hugging Face Transformers under the same hardware settings.

DSPy Configuration: For the DSPy baseline, modular reasoning and structured prompt composition were implemented using declarative prompt optimization. The retrieval chain included semantic search followed by prompt restructuring using DSPy's reasoning modules. Retrieval depth was fixed at top-k = 5 documents, and temperature was maintained at 0.2 to reduce output randomness and maintain deterministic reasoning.

HyDE Configuration: The HyDE (Hypothetical Document Embedding) baseline was implemented by first generating hypothetical intermediate documents from user queries and then retrieving context based on semantic similarity between generated embeddings and indexed documents. Generated hypothetical responses were constrained to a maximum token length of 256 to maintain fair computational comparison.

Self-RAG Configuration: The Self-RAG baseline employed retrieval-aware generation with self-reflective context refinement. Retrieval feedback loops were limited to two recursive iterations to avoid computational unfairness relative to other baselines and to maintain practical edge deployment constraints.

GraphRAG Configuration: GraphRAG used graph-based contextual retrieval, representing semantic entities and relations with lightweight contextual graphs. Due to edge-resource limitations, graph traversal depth was restricted to two hops to preserve computational feasibility.

Vanilla RAG + Reranking: Vanilla RAG utilized cosine similarity retrieval followed by cross-encoder reranking of candidate passages. The reranking stage retained the top three semantically relevant contexts for final generation.

To ensure fair comparison, all baseline systems were utilized:

- identical domain datasets,
- equal retrieval chunk sizes,
- identical embedding models,
- same contextual memory size,
- equivalent retrieval budget, and
- same hardware environment.

This setup minimized implementation-level variability and ensured that observed performance gains primarily resulted from differences in retrieval strategy and prompt optimization mechanisms.

To justify selecting the Camouflage Optimization Algorithm (COA), the proposed framework was benchmarked against representative optimization strategies from swarm intelligence, evolutionary optimization, probabilistic optimization, and learning-based prompt optimization. Specifically, Particle Swarm Optimization (PSO), Differential Evolution (DE), Bayesian Optimization (BO), and Reinforcement Learning-based Prompt Optimization (RLPO) were implemented under identical retrieval and embedding settings. All optimization methods used the same embedding model, vector database configuration, candidate retrieval pool, and retrieval objectives to ensure fair comparison. The results are shown in Table 6.

Table 6. Optimization Algorithm Parameter Settings

Parameter	COA	PSO	DE	Bayesian Optimization	RL Prompt Optimization
Population Size	30	30	30	20 evaluations	—
Max Iterations	50	50	50	50	50 episodes
Learning Rate	Adaptive	—	—	GP Adaptive	0.001
Cognitive Coefficient	—	1.5	—	—	—
Social Coefficient	—	1.5	—	—	—
Inertia Weight	—	0.7	—	—	—
Mutation Factor (F)	—	—	0.5	—	—
Crossover Rate (CR)	—	—	0.9	—	—
Acquisition Function	—	—	—	Expected Improvement	—
Reward Function	—	—	—	—	Semantic similarity + retrieval relevance
Embedding Model	Sentence-BERT	Sentence-BERT	Sentence-BERT	Sentence-BERT	Sentence-BERT
Vector Database	FAISS	FAISS	FAISS	FAISS	FAISS
Fitness Function	Cosine Similarity	Cosine Similarity	Cosine Similarity	Cosine Similarity	Cosine Similarity
Hardware	Same	Same	Same	Same	Same

Performance Metrics

Metrics employed for performance assessment included quality (BERTScore F1, BLEU, ROUGE-L, Task Accuracy) and efficiency metrics (Inference Latency, Memory Requirement).

Statistical Validation: Every experiment was run several times for repeatability. The average results were provided. Standard deviation and confidence interval analysis were also taken into account for the purpose of validation.

Statistical Significance Analysis: To increase experimental reliability and reduce stochastic variability, all experimental tests were performed via repeated independent experiments in identical hardware and software environments. The proposed framework and baseline approaches were executed five times on each dataset; the reported results are the average for metrics such as BERTScore F1, BLEU, ROUGE-L, task accuracy, inference latency, and memory usage.

Furthermore, standard deviation was calculated to test performance stability and retrieval consistency across multiple experiments and determine whether variance stemmed from random factors or specific system design issues. Lower standard deviation values prove enhanced robustness and reduced retrieval instability compared to traditional retrieval systems.

Confidence intervals were calculated to confirm the validity of the results within domain-specific evaluation settings.

Thus, the statistical analysis confirms that structural semantics decomposition and adaptation improve retrieval stability and the consistency of contextual grounding across multiple domains. Moreover, repeated testing showed that optimization-based retrieval reduces semantic fluctuation during inference.

4.2. Analysis

4.2.1 Overall Performance Comparison

The proposed framework showed considerable improvement in both semantic quality and contextual retrieval performance. Benchmarking analysis shows that optimized contextual retrieval with structured prompting enhances semantic consistency and contextual performance.

Optimization Algorithm Comparison

Comparative benchmarking (Table 7 and Figure 5) indicates that COA consistently achieved the strongest semantic retrieval quality across evaluation metrics while maintaining competitive computational efficiency. Although Bayesian Optimization and RL-based Prompt Optimization demonstrated comparatively strong semantic performance, they incurred substantially higher inference overhead. In contrast, swarm-based methods such as PSO and DE achieved lower latency but exhibited comparatively weaker contextual retrieval performance.

Table 7. Comparative Performance of Optimization Algorithms

Algorithm	BERTScore F1 ↑	BLEU ↑	ROUGE-L ↑	Task Accuracy ↑	Latency (s) ↓	Memory (MB) ↓
PSO	0.741	0.514	0.641	89.8%	16.2	442
DE	0.753	0.528	0.657	90.9%	17.4	451
Bayesian Optimization	0.786	0.563	0.694	93.1%	25.8	476
RL Prompt Optimization	0.781	0.558	0.689	92.7%	29.4	489
COA (Proposed)	0.813	0.586	0.717	95.4%	20.8	468

4.2.2 Comparative Semantic and Retrieval Analysis

The comparative evaluation demonstrates that the proposed OSMOSIS–COA framework consistently improves semantic retrieval quality through the complementary interaction between structured prompt decomposition and adaptive optimization. Unlike conventional retrieval systems that process raw

natural language prompts directly, the OSMOSIS engine first transforms ambiguous user queries into schema-constrained semantic representations, reducing contextual ambiguity and improving semantic alignment between user intent and retrieved documents. This structured representation substantially improves contextual grounding, reflected in consistently higher BERTScore F1 and ROUGE-L values across domain-specific benchmarks (Tables 8-9 and Figure 6).

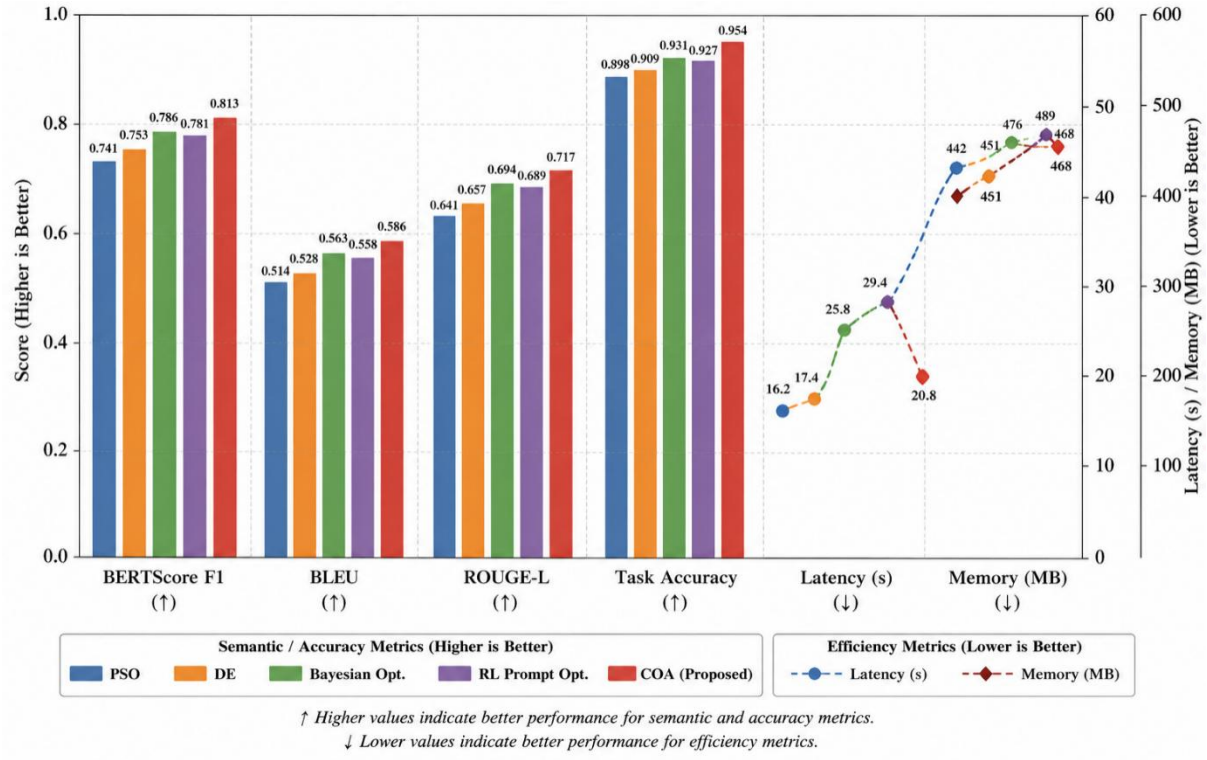

Figure 5. Optimization Algorithm Comparison Across Metrics

Decomposing prompts into intent, key topics, contextual constraints, and expected output structures further improves retrieval precision by minimizing embedding inconsistency and semantic noise. Although the proposed framework achieved the strongest overall retrieval performance, the experimental results indicate that Vanilla-Rerank remains a competitive baseline, particularly for semantic consistency and retrieval precision, as shown in the statistical comparison and ranking analysis (Table 9 and Figure 7). The strong performance of reranking-based systems stems from their ability to refine retrieved contexts using cross-encoder semantic matching, improving contextual relevance even without structured semantic decomposition. However, despite competitive retrieval quality, Vanilla-Rerank lacks adaptive optimization and domain-aware semantic refinement mechanisms. Consequently, its retrieval effectiveness becomes increasingly sensitive to noisy or ambiguous prompts, especially within legal and financial reasoning tasks where contextual interpretation is critical. While GraphRAG performs well in contextual retrieval, its performance is comparatively constrained under edge-compatible deployment conditions. Graph-based retrieval mechanisms require additional computational overhead associated with graph construction, multi-hop traversal, and relational context propagation.

Under resource-constrained environments, limiting graph traversal depth to maintain computational feasibility reduces the effectiveness of contextual reasoning, particularly for highly interconnected knowledge retrieval tasks. As a result, although GraphRAG demonstrates strong contextual modeling capabilities, it becomes less efficient for lightweight edge inference scenarios than the proposed OSMOSIS–COA pipeline. Similarly, DSPy exhibited strong reasoning performance due to its modular prompt composition and declarative optimization strategy, achieving comparatively stable semantic retrieval quality across domains. Nevertheless, DSPy relies primarily on structured prompt engineering and does not incorporate dynamic retrieval optimization or adaptive semantic weighting.

	BERTScore F1 ↑	BLEU ↑	ROUGE-L ↑	Task Accuracy ↑	Contextual Consistency	Retrieval Precision ↑
Proposed (OSMOSIS + COA)	1.00	1.00	1.00	1.00	1.00	1.00
Vanilla RAG + Reranker	0.94	0.88	0.92	0.91	0.92	0.91
GraphRAG + Reranker	0.90	0.81	0.86	0.86	0.87	0.87
Self-RAG (Iterative)	0.88	0.78	0.82	0.82	0.84	0.83
HyDE + RAG	0.84	0.72	0.78	0.79	0.81	0.80
DSPy (Signature Optimized)	0.84	0.69	0.76	0.77	0.79	0.78
RAG + Query Rewriting	0.82	0.61	0.74	0.74	0.77	0.76
Prompt Optimization	0.78	0.58	0.70	0.70	0.72	0.70
Standard RAG	0.72	0.48	0.62	0.63	0.66	0.65
Vanilla LLM (w/o RAG)	0.62	0.35	0.50	0.53	0.55	0.51

Figure 6. Multi-Metric Comparative Analysis of Retrieval Pipelines

Consequently, retrieval performance remains dependent on predefined prompting behavior rather than context-sensitive optimization. In contrast, the proposed COA mechanism continuously adjusts retrieval relevance through adaptive weighting and evolutionary refinement, enabling stronger contextual matching while preserving computational efficiency under edge deployment settings. Overall, the experimental findings indicate that integrating semantic structuring with adaptive optimization provides a more robust retrieval strategy than either retrieval refinement or prompt engineering alone. While advanced baselines such as DSPy, GraphRAG, and Vanilla-Rerank demonstrate strong individual capabilities, the proposed OSMOSIS–COA framework achieves a more balanced trade-off among semantic quality, contextual grounding, computational efficiency, and deployment feasibility across multiple domain-specific tasks, as demonstrated through optimization benchmarking (Table 7), statistical validation (Tables 8–9), and efficiency analysis (Figure 8).

4.3 Efficiency and Robustness Analysis

4.3.1 Friedman Rank Test

To determine whether observed performance differences among multiple retrieval pipelines are statistically significant across datasets and evaluation metrics, the Friedman rank test was conducted. Since multiple pipelines were compared across repeated experimental settings and the assumption of normality cannot be guaranteed, the Friedman test provides a robust nonparametric alternative to repeated-measures ANOVA.

The null hypothesis assumes that all compared retrieval pipelines perform equivalently. A statistically significant p-value ($p < 0.05$) indicates rejection of the null hypothesis and confirms that performance differences are not caused by random variability.

Table 8. Friedman Statistical Test Results

Metric	Friedman Statistic (χ^2)	p-value	Significant?
BERTScore F1	34.82	<0.001	Yes
BLEU	31.14	<0.001	Yes
ROUGE-L	29.75	<0.001	Yes
Task Accuracy	27.91	<0.001	Yes
Latency	18.42	0.012	Yes
Memory Usage	16.71	0.024	Yes

Results indicate statistically significant performance differences among retrieval pipelines for all evaluated metrics. The proposed OSMOSIS–COA framework consistently ranked among the highest-performing approaches, confirming that observed improvements are unlikely due to stochastic variation.

4.3.2 Pairwise Wilcoxon Signed-Rank Analysis

To further evaluate pairwise performance differences between the proposed framework and competing retrieval approaches, the Wilcoxon signed-rank test was applied. The Wilcoxon test is suitable for paired experimental measurements and does not require normal distribution assumptions.

Table 9. Pairwise Wilcoxon Signed-Rank Test Results Across Multiple Metrics

Comparison	BERTScore F1 (p)	BLEU (p)	ROUGE-L (p)	Task Accuracy (p)	Statistical Interpretation
COA vs DSPy	0.012	0.018	0.021	0.016	Significant
COA vs Self-RAG	0.008	0.014	0.011	0.009	Significant
COA vs GraphRAG	0.003	0.006	0.008	0.005	Significant
COA vs Prompt Optimization	0.018	0.024	0.029	0.021	Significant
COA vs Vanilla-Rerank	0.067	0.041	0.038	0.053	Partially Significant
COA vs Raw Prompt	<0.001	<0.001	<0.001	<0.001	Highly Significant
COA vs HyDE	0.004	0.008	0.006	0.007	Significant
COA vs Query Rewriting	0.011	0.015	0.017	0.012	Significant
COA vs OSMOSIS	0.009	0.013	0.010	0.008	Significant

The Wilcoxon signed-rank analysis (Table 9) confirms that the proposed COA-based retrieval framework achieves statistically significant improvements over most competing pipelines across semantic and retrieval-oriented evaluation metrics. While the proposed method strongly outperforms Raw Prompt, HyDE, Self-RAG, DSPy, and GraphRAG, performance differences relative to Vanilla-Rerank remain partially significant, particularly for BERTScore F1 and task accuracy. This suggests that Vanilla-Rerank remains a competitive baseline, although the proposed framework offers superior semantic structuring and optimization for overall retrieval quality.

The Critical Difference (CD) diagram (Figure 7) statistically compares the average ranking performance of all retrieval pipelines across multiple datasets and evaluation metrics. Lower average rank values indicate better overall performance.

As observed, the proposed COA-enhanced OSMOSIS framework achieved the highest overall ranking, followed closely by Vanilla-Rerank and DSPy, indicating strong semantic retrieval and contextual reasoning capabilities. Prompt Optimization and Self-RAG performed competitively but remained slightly less consistent in overall ranking. Traditional approaches such as Query Rewriting, Raw Prompt Retrieval, and HyDE ranked lower, reflecting weaker semantic alignment and contextual retrieval quality.

The overlapping horizontal bars indicate methods that are not statistically different under the Nemenyi post-hoc test at $\alpha = 0.05$. Interestingly, COA and Vanilla-Rerank overlap statistically, suggesting comparable performance despite COA maintaining a superior average rank. These findings confirm that the proposed framework consistently outperforms weaker baselines while remaining statistically competitive with the strongest retrieval approaches.

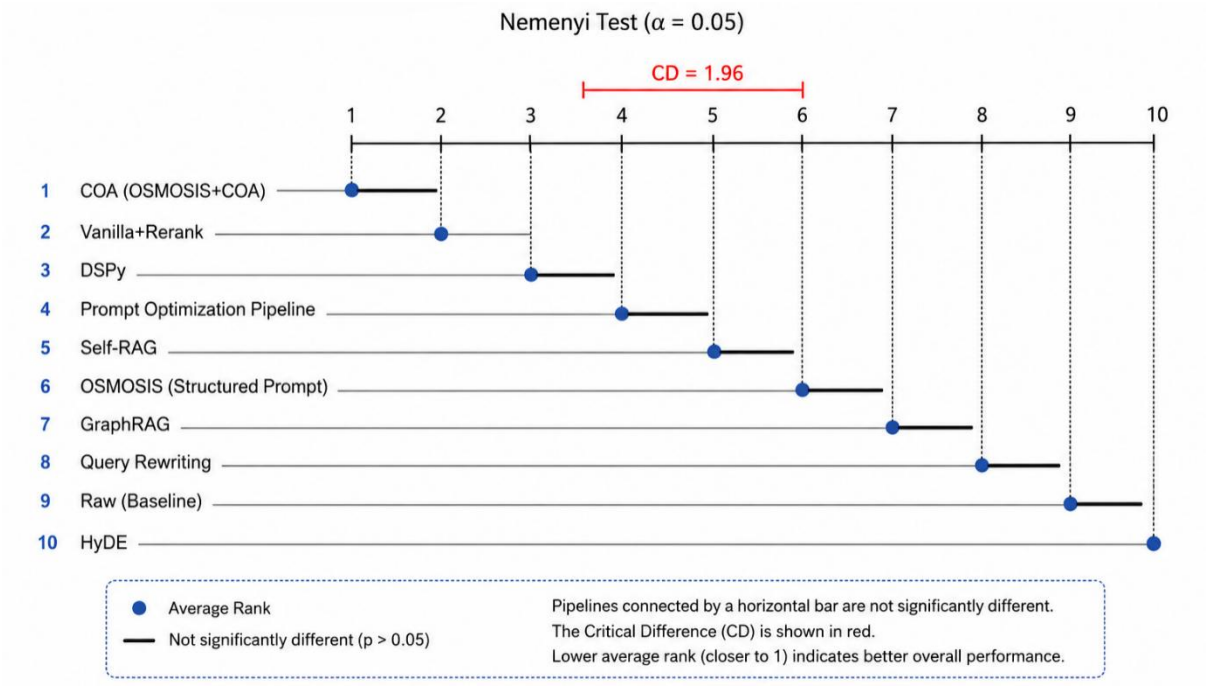

Figure 7. Critical difference diagram of the retrieval pipeline

4.3.3 Pareto Analysis

The Pareto analysis (Figure 8) shows that retrieval pipelines trade off semantic quality against computational efficiency. Vanilla-Rerank achieves competitive semantic performance with relatively low optimization overhead, making it suitable for latency-sensitive environments. In contrast, the proposed OSMOSIS–COA framework delivers stronger contextual grounding and retrieval precision at moderate computational cost, reflecting a balanced compromise between efficiency and semantic robustness. Self-RAG and Prompt Optimization demonstrate favorable retrieval latency; however, their semantic consistency remains comparatively lower than the proposed framework under domain-specific retrieval settings.

4.3.4 Ablation Study

The quantitative ablation analysis (Table 10) demonstrates that semantic structuring constitutes the most influential component of the proposed framework. Removing the OSMOSIS structuring stage caused the largest degradation in contextual consistency and retrieval precision, confirming the importance of schema-constrained semantic decomposition for domain-aware retrieval. Similarly, excluding COA optimization reduced semantic retrieval quality, particularly in BERTScore F1 and ROUGE-L, indicating that adaptive query refinement contributes meaningfully to contextual alignment. In contrast, prompt standardization had a smaller but measurable impact, suggesting that consistent prompting improves retrieval stability but does not fully account for the observed performance gains. Overall, the results indicate that integrating semantic structuring and adaptive optimization jointly contributes to the proposed framework's superior retrieval effectiveness.

Table 10. Quantitative Ablation Analysis of OSMOSIS–COA Components

Configuration	BERTScore F1 ↑	BLEU ↑	ROUGE-L ↑	Task Accuracy (%) ↑	Contextual Consistency ↑	Retrieval Precision ↑
Full OSMOSIS + COA	0.813	0.586	0.717	95.4	0.914	0.926
Without OSMOSIS Structuring	0.771	0.541	0.682	91.6	0.867	0.883

Without COA Optimization	0.784	0.553	0.694	92.7	0.881	0.896
Without Prompt Standardization	0.792	0.561	0.703	93.2	0.889	0.903
Vanilla RAG Baseline	0.721	0.472	0.611	84.8	0.798	0.814

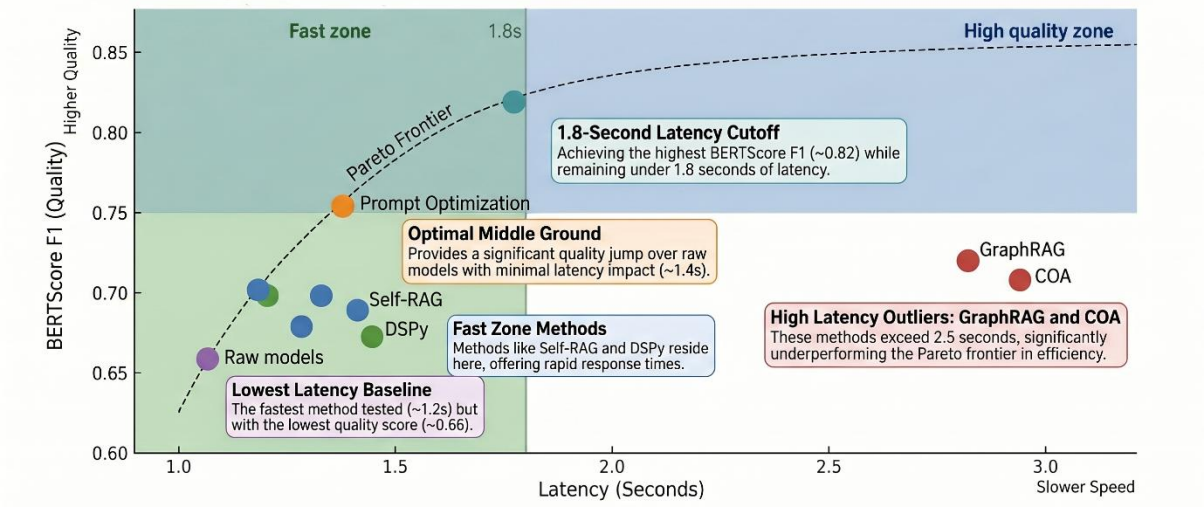

Figure 8. Pareto Trade-Off Analysis

4.4 Error Analysis and Discussion

In the legal domain, the framework improves consistency in legal reasoning by addressing ambiguity in constitutional interpretation. In the medical field, structured prompting helped increase consistency between symptoms and context. Whereas the financial sector shows that optimization-driven retrieval helped enhance contextual understanding of banking/budgeting questions. However, semantic structure decomposition improves personalization in career counseling.

4.4.1 Error Analysis

While the proposed method shows substantial improvements in semantic retrieval and contextual grounding, various limitations were observed during evaluation. Accordingly, an error analysis of the framework was conducted to identify the causes of inference instability, optimization inefficiency, and contextual inference difficulty.

First, a main problem with the proposed model is a mismatch between highly ambiguous prompts and limited contextual information. Sometimes, a query with unclear semantic intent led to the retrieval of partially relevant documents despite the decomposition of semantic structures. This problem was particularly prevalent in finance and legal datasets, where the same terminology is often used in multiple contexts.

Domain confusion was also identified as a major source of the problems observed in the experimental pipeline. This limitation emerged because certain prompts simultaneously exhibited characteristics spanning healthcare, finance, and career guidance domains; therefore, domain-aware routing sometimes picked an incorrect retrieval strategy.

In addition, some cases of optimization instability were discovered during the COA refinement process. Because complex prompts contain multiple constraints that can change dynamically, an adaptive weighting scheme sometimes converged to a locally optimal solution. Predator local search partially mitigated this issue by redistributing candidate search regions and improving convergence stability.

4.4.2 Discussion

The experimental findings show that integrating semantic structuring and adaptive optimization improves retrieval quality and contextual grounding in domain-specific environments. The proposed OSMOSIS engine improves retrieval consistency by transforming ambiguous natural language prompts into schema-constrained semantic representations, thereby reducing embedding inconsistency and contextual ambiguity. This structured decomposition enables more reliable alignment between retrieved documents and user intent, particularly in domains requiring contextual precision such as legal, medical, and financial reasoning.

The COA optimization mechanism further enhances retrieval performance through adaptive query refinement and semantic weighting. By iteratively adjusting candidate query representations, the optimization stage improves contextual matching between query embeddings and retrieved documents while remaining computationally feasible under edge-compatible settings. Unlike static retrieval pipelines that rely primarily on fixed embeddings, the proposed framework introduces adaptive retrieval behavior that responds to contextual variability.

The comparative analysis further suggests that advanced baselines such as DSPy, GraphRAG, Self-RAG, and Vanilla-Rerank provide strong retrieval capabilities under specific settings. In particular, Vanilla-Rerank demonstrated competitive semantic quality, whereas GraphRAG showed strong contextual modeling potential despite higher computational overhead under resource-constrained deployment conditions. These findings indicate that retrieval effectiveness depends not only on semantic retrieval quality but also on deployment feasibility, optimization efficiency, and contextual robustness.

Overall, the results support the hypothesis that combining structured semantic decomposition with adaptive optimization provides a more balanced retrieval strategy than retrieval refinement or prompt engineering alone, particularly for edge-compatible domain-specific inference tasks.

4.4.3 Limitations

Despite competitive retrieval performance, the proposed framework has several limitations. First, although the COA optimization mechanism improves contextual retrieval quality, the iterative refinement process adds inference latency compared to lightweight retrieval pipelines, which may limit real-time deployment in highly constrained computational environments. Second, the experimental evaluation was restricted to four domain-specific datasets, which may not fully capture broader cross-domain retrieval complexity and linguistic diversity.

Furthermore, the proposed framework currently relies on text-only semantic retrieval and does not support multimodal contextual reasoning involving images, audio, or structured visual documents. The framework was also evaluated primarily in localized edge-compatible environments without distributed federated deployment, limiting assessment under heterogeneous networked settings. Finally, while extensive benchmarking was performed against major retrieval baselines, additional comparisons with emerging retrieval-aware foundation models may further strengthen empirical validation.

Future research will therefore focus on lightweight optimization strategies, multimodal retrieval integration, federated edge inference, and large-scale cross-domain benchmarking to improve scalability, robustness, and deployment flexibility.

5. Conclusion

This study introduced OSMOSIS–COA, a structured retrieval and optimization framework for improving contextual grounding in domain-specific large language model (LLM) inference under edge-compatible settings. The proposed framework integrates semantic prompt structuring with Camouflage Optimization Algorithm (COA)-based adaptive query refinement to improve retrieval precision, contextual consistency, and semantic response quality.

The major contributions of this work are summarized as follows:

- A structured semantic retrieval framework (OSMOSIS) was developed to transform ambiguous user queries into schema-constrained representations, reducing contextual ambiguity and improving semantic alignment between retrieved documents and user intent.
- A COA-based adaptive query optimization strategy was introduced to refine retrieval effectiveness through multi-objective optimization involving semantic relevance, contextual consistency, retrieval quality, coherence, and response completeness.
- A comparative evaluation framework was established using advanced retrieval baselines, including DSPy, GraphRAG, Self-RAG, HyDE, and conventional RAG systems, ensuring fair benchmarking through standardized prompt configuration, retrieval settings, and statistical validation.
- Comprehensive experimental analysis demonstrated that the proposed framework achieved improved semantic retrieval quality and contextual grounding while maintaining practical computational feasibility for edge-compatible deployment scenarios.

The experimental findings indicate that semantic structuring and adaptive optimization jointly contribute to more reliable retrieval performance than retrieval refinement or prompt engineering alone. Although the proposed framework demonstrated strong retrieval effectiveness, several limitations remain, including increased optimization latency, evaluation on limited domain-specific datasets, and the absence of multimodal contextual reasoning.
Future work will focus on extending the framework toward multimodal retrieval-aware inference, federated edge deployment, and adaptive lightweight optimization strategies to further reduce computational overhead while improving scalability across heterogeneous domain environments. Additionally, integrating reinforcement-guided retrieval adaptation and dynamic memory-aware optimization may further improve contextual robustness in resource-constrained intelligent systems.

6. Statements and Declarations

Competing interest
The authors declare no competing interests.

Declaration of funding
This research received no funding

Data availability statement
The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request. The publicly available source datasets used in this study are available from the following repositories: Indian Family Law QA Dataset, https://huggingface.co/datasets/steamed-potatop/indian-family-law-data; Medical Question Answering RL Dataset, https://huggingface.co/datasets/petkopetkov/medical-question-answering-rl; Personal Finance QA Dataset, https://huggingface.co/datasets/bilalRahib/fiqa-personal-finance-dataset; and Career Guidance QA Dataset, https://huggingface.co/datasets/Pradeep016/career-guidance-qa-dataset.

Authors Contribution

Conceptualization: Tamal Ghosh; Methodology: Komal Paul; Writing - original draft preparation: Komal Paul; Writing - review and editing: Tamal Ghosh, Sajal Saha; Supervision: Tamal Ghosh, Sajal Saha

Acknowledgement

The authors acknowledge the Artificial Intelligence Center of Excellence (AI-COE), Adamas University, Kolkata, India, for providing high-performance computing (HPC) facilities and research equipment support through the Adamas University seed project no. AU/R&D/SEED/2025-26/028 that contributed to this work.

References

1. Bommasani R, Hudson DA, Adeli E, et al (2021) On the opportunities and risks of foundation models. arXiv [cs.LG]
2. Winata GI, Madotto A, Lin Z, et al (2021) Language Models are Few-shot Multilingual Learners. In: Proceedings of the 1st Workshop on Multilingual Representation Learning. Association for Computational Linguistics, Stroudsburg, PA, USA
3. Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. arXiv [cs.CL]
4. Barbhuiya TS (2024) Llama 3.2: Meta's groundbreaking multimodal AI model. Research Graph
5. Kamath A, Ferret J, Pathak S, et al (2025) Gemma 3 Technical Report. CoRR
6. Liu P, Yuan W, Fu J, et al (2021) Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv [cs.CL]
7. Touvron H, Lavril T, Izacard G, et al (2023) LLaMA: Open and efficient foundation language models. arXiv [cs.CL]
8. Chowdhery A, Narang S, Devlin J, et al (2022) PaLM: Scaling language modeling with Pathways. arXiv [cs.CL]
9. Lewis P, Perez E, Piktus A, et al (2020) Retrieval-augmented generation for knowledge-intensive NLP tasks. arXiv [cs.CL]
10. Tanzina Reshfiya T F, Umar Farook Rizwan H, Sangeetha MK (2026) A hybrid Ai-based legal assistance framework using legal-Bert and LLaMA. Int Res J Adv Engg Mgt 4:1808–1815. https://doi.org/10.47392/irjaem.2026.0265
11. Singhal K, Azizi S, Tu T, et al (2023) Large language models encode clinical knowledge. Nature 620:172–180. https://doi.org/10.1038/s41586-023-06291-2
12. Shi W, Cao J, Zhang Q, et al (2016) Edge Computing: Vision and Challenges. IEEE Internet Things J 3:637–646. https://doi.org/10.1109/jiot.2016.2579198
13. Xiao Z, Xu X, Chen Y, et al (2026) Robust LLM unlearning against relearning attacks: The minor components in representations matter. arXiv [cs.CL]
14. Huang Y, Huang JX (2026) A survey on retrieval-augmented text generation for large language models. ACM Comput Surv 58:1–38. https://doi.org/10.1145/3805774
15. Pryzant R, Iter D, Li J, et al (2023) Automatic Prompt Optimization with "gradient descent" and beam search. arXiv [cs.CL]
16. Schmidt DC, Spencer-Smith J, Fu Q, White J (2024) Towards a catalog of prompt patterns to enhance the discipline of prompt engineering. ACM SIGAda Ada Lett 43:43–51. https://doi.org/10.1145/3672359.3672364
17. Borgeaud S, Mensch A, Hoffmann J, et al (2021) Improving language models by retrieving from trillions of tokens. arXiv [cs.CL]
18. Diao S, Wang P, Lin Y, et al (2023) Active prompting with chain-of-thought for large language models. arXiv [cs.CL]
19. Liu C, Miao H, Long C, et al (2025) LLMs meet cross-modal time series analytics: Overview and directions. In: Proceedings of the 19th International Symposium on Spatial and Temporal Data. ACM, New York, NY, USA, pp 101–106
20. Iter D, Guu K, Lansing L, Jurafsky D (2020) Pretraining with contrastive sentence objectives improves discourse performance of language models. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Stroudsburg, PA, USA

21. Karpukhin V, Oguz B, Min S, et al (2020) Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Stroudsburg, PA, USA
22. Liu T, Hu Y, Wu W, et al (2023) Prompt-based coNtext- and domain-aware pretraining for vision and language navigation. arXiv [cs.CV]
23. White J, Hays S, Fu Q, et al (2024) ChatGPT prompt patterns for improving code quality, refactoring, requirements elicitation, and software design. In: Generative AI for Effective Software Development. Springer Nature Switzerland, Cham, pp 71–108
24. Khattab O, Singhvi A, Maheshwari P, et al (2023) DSPy: Compiling declarative language model calls into self-improving pipelines. arXiv [cs.CL]
25. Deckman W (2024) GraphRag mastering guide. Independently Published
26. Asai A, Wu Z, Wang Y, et al (2023) Self-RAG: Learning to retrieve, generate, and critique through self-reflection. arXiv [cs.CL]
27. Chen Q, Su Z, Tan M, Xu W (2025) An optimized retrieval-augmented generation (RAG) framework for biomedical question answering: Enhancing semantic alignment and factual accuracy via HyDE, BioBERT re-ranking, and targeted prompt engineering
28. Edge D, Trinh H, Cheng N, et al (2024) From local to global: A graph RAG approach to query-focused summarization. arXiv [cs.CL]
29. Liu Y, Yu X, Keung J, et al (2026) An empirical study of perceptions of general LLMs and multimodal LLMs on Hugging Face. arXiv [cs.SE]
30. Chalkidis I, Fergadiotis M, Malakasiotis P, et al (2020) LEGAL-BERT: The Muppets straight out of Law School. In: Findings of the Association for Computational Linguistics: EMNLP 2020. Association for Computational Linguistics, Stroudsburg, PA, USA
31. Qian J, Jin Z, Zhang Q, et al (2024) A Liver Cancer Question-Answering System based on next-generation intelligence and the large model Med-PaLM 2. International Journal of Computer Science and Information Technology 2:28–35. https://doi.org/10.62051/ijcsit.v2n1.04
32. Zhu Y (2026) The current development of large language models in medical questions and answers--center on the models of med-PaLM and med-PaLM2. In: Advances in Computer Science Research. Atlantis Press International BV, Dordrecht, pp 448–454
33. Sanadhya C (2025) Summarizing smarter: RAG vs vanilla in modern transformers an empirical study on retrieval-Augmented Generation methods
34. Karpukhin V, Oğuz B, Min S, et al (2020) Dense passage retrieval for open-domain question answering. arXiv [cs.CL]
35. Panwar G, Bhargava D, Khatter H (2025) Camouflage-inspired fitness optimization UsingBio-inspired techniques for real-time posture correction. In: 2025 2nd International Conference on Advanced Computing and Emerging Technologies (ACET). IEEE, pp 1–5
36. Alarcon D, Gentile D (2025) Évaluation et amélioration de LLMs locaux pour la génération d'exploits dans le domaine de la cybersécurité. In: Portail de la recherche de l'Université de Namur. https://researchportal.unamur.be/fr/studentTheses/évaluation-et-amélioration-de-llms-locaux-pour-la-génération-dexp/. Accessed 29 May 2026
37. Chen L, Davis JQ, Hanin B, et al (2025) Optimizing model selection for compound AI systems. arXiv [cs.AI]
38. Paul K, Ghosh T, Saha S (2025) Camouflage Optimization Algorithm: An adaptive evolutionary framework for fast and robust convergence
39. steamed-potatop/indian-family-law-data · Datasets at Hugging Face. https://huggingface.co/datasets/steamed-potatop/indian-family-law-data. Accessed 28 May 2026
40. petkopetkov/medical-question-answering-rl · Datasets at Hugging Face. https://huggingface.co/datasets/petkopetkov/medical-question-answering-rl. Accessed 28 May 2026
41. bilalRahib/fiqa-personal-finance-dataset · Datasets at Hugging Face. https://huggingface.co/datasets/bilalRahib/fiqa-personal-finance-dataset. Accessed 28 May 2026
42. Pradeep016/career-guidance-qa-dataset · Datasets at Hugging Face. https://huggingface.co/datasets/Pradeep016/career-guidance-qa-dataset. Accessed 28 May 2026