PathwiseRAG: Multi-Dimensional Exploration and Integration Framework

Hengrui Zhang[1], Pin-Siang Huang[2], Zhen Zhang[1], Peican Lin[1], Yao-Ching Yu[2], Bo Hu[1], Yulu Du[1*]

[1]Chongqing University of Posts and Telecommunications
[2]National Taiwan University of Science and Technology

*Correspondence: duyl@cqupt.edu.cn

Abstract

Conventional retrieval-augmented generation (RAG) systems employ rigid retrieval strategies that create: (1) knowledge blind spots across domain boundaries, (2) reasoning fragmentation when processing interdependent concepts, and (3) contradictions from conflicting evidence sources. Motivated by these limitations, the paper introduces PathwiseRAG, which addresses these challenges through: intent-aware strategy selection to eliminate blind spots, dynamic reasoning networks that capture sub-problem interdependencies to overcome fragmentation, and parallel path exploration with adaptive refinement to resolve conflicts. The framework models query intent across semantic and reasoning dimensions, constructs a directed acyclic graph of interconnected sub-problems, and explores multiple reasoning trajectories while continuously adapting to emerging evidence. Evaluation across five challenging benchmarks spanning single-hop to multi-hop reasoning demonstrates significant improvements over state-of-the-art RAG systems, with average accuracy gains of 4.9% and up to 6.9% on complex queries, establishing a new paradigm for knowledge-intensive reasoning by transforming static retrieval into dynamic, multi-dimensional exploration.

1 Introduction

Large language models have transformed natural language processing yet struggle with knowledge-intensive tasks requiring factual precision and multi-step reasoning. Retrieval-augmented generation (RAG) addresses these limitations by incorporating external knowledge. Existing RAG systems operate through static pipelines that create three critical limitations. First, static retrieval produces knowledge blind spots, missing crucial connections between quantum algorithms and encryption vulnerabilities. Second, conventional RAG cannot model interdependent concepts across documents, processing mortgage securities, default swaps, and regulations in isolation. Third, these systems cannot adapt retrieval based on intermediate findings, lacking the dynamic exploration required for complex knowledge tasks. Previous improvements still treat information acquisition as linear rather than the dynamic, branching exploration needed for complex domains.

PathwiseRAG reconceptualizes RAG as dynamic, multi-dimensional exploration through: (1) dual-stream intent analysis modeling semantic content and reasoning requirements for strategy selection; (2) reasoning network construction organizing interdependent sub-problems as directed acyclic graphs; and (3) parallel path exploration continuously adjusting networks as evidence emerges.

Evaluation on five datasets spanning different reasoning complexities (HotpotQA, StrategyQA, ComplexWebQuestions, Natural Questions, and TriviaQA) demonstrates substantial improvements over state-of-the-art RAG systems, with average accuracy gains of 4.9% and up to 6.9% on complex queries. Ablation studies confirm each component's critical contribution.

The key contributions include: (1) intent-aware strategy selection through dual-stream analysis jointly modeling semantic and reasoning dimensions; (2) dynamic reasoning networks representing query decomposition as evolving DAG structures with real-time adjustment; and (3) parallel path exploration coordinating multiple retrieval strategies with conflict resolution through weighted evidence integration.

2 Related Work

2.1 Retrieval-Augmented Generation

Traditional RAG systems primarily utilize text-based retrieval (Lewis et al., 2020; Izacard and Grave, 2021), but recent approaches have expanded to multimodal data sources. Liu et al. (Liu et al.,

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 22916–22937
November 4-9, 2025

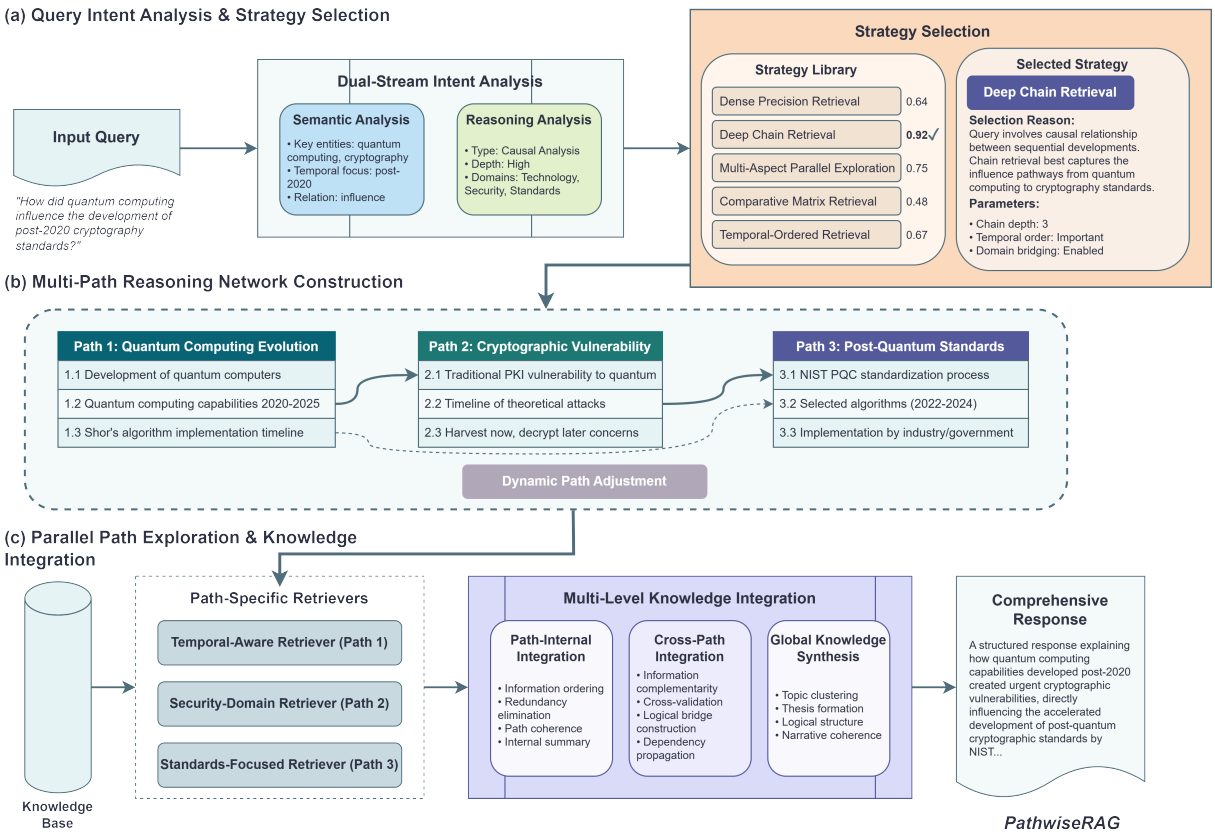

Figure 1: PathwiseRAG: multi-dimensional exploration and integration framework. (a) query-intent analysis & strategy selection, (b) multi-path reasoning network construction, (c) parallel path exploration with multi-level knowledge integration.

2025) introduced a hierarchical multi-agent framework for heterogeneous data sources, while Gupta et al. (Gupta et al., 2024) highlighted limitations of single-source retrieval for complex queries. Graph-based approaches like LightRAG (Guo et al., 2024) and advanced graph structures (Dong et al., 2024) enhance textual interdependencies but often sacrifice fine-grained details that PathwiseRAG preserves.

2.2 Multi-Agent Frameworks

Agent-based RAG architectures improve system modularity and query processing (Jeong, 2024; Han et al., 2025). Chan et al. (2024) (Chan et al., 2024) focused on query refinement to improve retrieval quality, while Su et al. (2024) (Su et al., 2024) developed a system for real-time information needs. PathwiseRAG extends these frameworks through coordinated reasoning paths and intent-driven strategy selection that existing approaches typically lack.

2.3 Knowledge Integration

Recent works have developed specialized techniques for knowledge integration. Mavromatidis and Karypis (2024) (Mavromatids and Karypis, 2024) introduced GNN-RAG for enhanced reasoning, while Wu et al. (2024) (Wu et al., 2024) developed a domain-specific graph RAG for medicine. For multimodal data, Xia et al. (2024) (Xia et al., 2024) created a versatile RAG system for medical vision-language models, and Edge et al. (2024) (Edge et al., 2024) proposed a GraphRAG approach transitioning from local to global integration. PathwiseRAG advances beyond these fixed integration strategies through its adaptive three-level process.

2.4 Adaptive Exploration

Adaptive exploration strategies have emerged as critical for complex information retrieval. Procko and Ochoa (2024) (Procko and Ochoa, 2024) highlighted the benefits of topological relationships for document modeling, while Su et al. (2024) (Su et al., 2024) and Toro et al. (2024) (Toro et al., 2024) introduced dynamic RAG systems that adapt

retrieval patterns based on emerging needs. PathwiseRAG extends these approaches through comprehensive intent analysis and parallel path exploration that existing methods lack.

3 Methodology

3.1 Framework Overview

Algorithm 1 PathwiseRAG Algorithm

Require: Query q, document corpus $\mathcal{D}$, strategy set $\mathcal{S}$
Ensure: Generated answer A
$I_q \leftarrow \text{IntentAnalysis}(q)$
$s^* \leftarrow \arg\max_{s \in \mathcal{S}} S(s, I_q)$
$G_0 = (V_0, E_0, \omega_0) \leftarrow \text{Network}(q, I_q, s^*)$
$P_0 = \{p_1, p_2, ..., p_k\} \leftarrow \text{GeneratePaths}(G_0)$
for $t = 0$ to $T - 1$ **do**
  Execute in parallel for all $p_i \in P_t$:
  $D_i^t \leftarrow \text{Retrieve}(p_i, \mathcal{D}, \theta_{s^*})$
  $K_i^t \leftarrow \phi_{\text{internal}}(D_i^t)$
  $K^t \leftarrow \phi_{\text{cross}}(\{K_i^t\}_{i=1}^k)$
  $G_{t+1} \leftarrow \Phi(G_t, K^t, \gamma)$
  $P_{t+1} \leftarrow \delta(P_t, G_{t+1})$
end for
$K \leftarrow \phi_{\text{global}}(\{K^t\}_{t=0}^{T-1})$
$A \leftarrow \text{Generate}(q, K)$
return A

PathwiseRAG implements a three-stage exploration pipeline (Fig. 1) for complex query processing. Algorithm 1 outlines the core execution flow:

First, dual-stream intent analysis extracts representation I_q from query q, capturing both semantic elements (entities, relations) and reasoning requirements (inference type τ, depth δ, domains $\mathbf{D}$). This representation guides optimal strategy selection s^* from strategy set $\mathcal{S}$.

Second, a directed acyclic graph $G_0 = (V_0, E_0, \omega_0)$ structures the reasoning process, where V_0 represents subproblems, E_0 represents dependencies, and ω_0 assigns priority weights. This network decomposes complex queries into k parallel reasoning paths $P_0 = \{p_1, p_2, ..., p_k\}$.

Third, during T iterations, each path p_i retrieves relevant documents D_i^t using strategy parameters θ_{s^*}. Retrieved information undergoes hierarchical integration: path-internal (ϕ_{internal}), cross-path (ϕ_{cross}), and global synthesis (ϕ_{global}). The reasoning network dynamically updates ($G_{t+1} = \Phi(G_t, K^t, \gamma)$) based on discovered information, where γ controls adjustment frequency, enabling adaptive exploration of complex information spaces.

3.2 Query Intent Analysis and Strategy Selection

Query intent analysis forms the foundation of PathwiseRAG, enabling retrieval strategies tailored to underlying information needs. As illustrated in Fig. 6, the study implements a dual-stream neural architecture (detailed in Appendix A.2) that processes queries through parallel computational pathways:

$$\text{Intent}(q) = \mathcal{F}_{\text{concat}}[\text{Semantic}(q), \text{Reasoning}(q)] \tag{1}$$

The semantic analysis extracts and combines three key elements to capture the query's informational context:

$$\text{SemanticAnalysis}(q) = \{\mathbf{E}, \mathbf{R}, \mathbf{T}\} \tag{2}$$

where $\mathbf{E} = \{e_1, e_2, ..., e_n\}$ represents key entities extracted through attention-weighted token classification, $\mathbf{R} = \{r_1, r_2, ..., r_m\}$ contains relation triples $(e_i, r_{\text{type}}, e_j)$ identified via pairwise classification over entity combinations, and $\mathbf{T}$ captures temporal markers through specialized token detection. These elements collectively form a structured semantic representation of the query's content.

The reasoning analysis identifies the query's logical structure through pattern recognition:

$$\text{ReasoningAnalysis}(q) = \{\tau, \delta, \mathbf{D}\} \tag{3}$$

The reasoning type τ is classified into one of {causal, comparative, procedural, hypothetical, factual}, depth δ is assessed as {low, medium, high}, and domains $\mathbf{D} = \{d_1, d_2, ..., d_k\}$ are identified through multi-label classification. These elements together characterize the reasoning requirements of the query.

The information need graph $G_{\text{info}} = (\mathbf{V}, \mathbf{E}', \mathbf{C})$ structures query elements into a coherent representation. This graph is constructed by first mapping extracted entities to nodes $\mathbf{V}$, connecting them with edges $\mathbf{E}'$ based on identified relations, and organizing them into domain clusters $\mathbf{C}$ using domain classification. The graph is further enriched with implicit relationships from external knowledge bases and transitive closures to capture logical connections not explicitly stated in the query.

The optimal retrieval strategy is selected by scoring candidate strategies against query intent:

$$S(s_k, q) = \alpha\, M_I(s_k, I_q) + \beta\, M_B(s_k, B_q) + \gamma\, M_D(s_k, D_q) + \varepsilon\, M_C(s_k, C_q). \quad (4)$$

Each component metric (M_I, M_B, M_D, M_C) evaluates a specific aspect of strategy-query alignment, including intent matching, breadth compatibility, depth alignment, and critical aspect coverage, respectively. Detailed formulations and theoretical justifications for these metrics are provided in Appendix A.3.

The strategy with the highest score is selected: $s^* = \arg\max_{s_k \in \mathbf{S}} S(s_k, q)$, where $\mathbf{S}$ is the set of available strategies including Dense Precision Retrieval, Deep Chain Retrieval, Multi-Aspect Parallel Exploration, Comparative Matrix Retrieval, and Temporal-Ordered Retrieval as detailed in Appendix A.2. For the selected strategy, specific retrieval parameters θ_{s^*} are generated to control execution dynamics.

3.3 Multi-Path Reasoning Network Construction

Following strategy selection, PathwiseRAG constructs a structured reasoning network to guide multi-path exploration. The subproblem decomposition process is defined as:

$$\mathcal{D} : q \mapsto \mathcal{S}(q) = \{s_1, s_2, \ldots, s_n\} \quad (5)$$

Each subproblem $s_i = (c_i, t_i, r_i)$ contains content focus c_i, type t_i, and retrieval approach r_i. The decomposition uses a trained classifier:

$$\mathcal{S}(q) = \{s_i | \mathcal{C}_{\text{decomp}}(q, i) > \tau_{\text{decomp}}, 1 \le i \le m\} \quad (6)$$

where $\mathcal{C}_{\text{decomp}}(q, i)$ computes the probability that position i in query q represents a logical breakpoint for subproblem decomposition, and τ_{decomp} is a confidence threshold.

These subproblems form a directed acyclic graph $G = (V, E, \omega)$, where V corresponds to subproblems, E indicates dependencies, and $\omega : V \to \mathbb{R}^+$ assigns priority scores. The priority score $\omega(v_i)$ combines relevance to the query, estimated difficulty, and predicted information gain.

The edges representing dependencies are determined by:

$$E = \{(v_i, v_j) | \mathcal{D}_{\text{dep}}(s_i, s_j) > \tau_{\text{dep}}, i \neq j\} \quad (7)$$

where $\mathcal{D}_{\text{dep}}(s_i, s_j)$ evaluates whether resolving subproblem s_i is logically prerequisite to addressing s_j, and τ_{dep} is the dependency threshold.

Multiple reasoning paths are extracted from this graph, with each path defined as:

$$P_i = \{q \to v_{i,1} \to v_{i,2} \to \cdots \to v_{i,k_i}\} \quad (8)$$

where $v_{i,j}$ is the j-th node in path i, and k_i is the path length. Paths are extracted using a modified search algorithm that prioritizes nodes with higher ω values while ensuring path diversity.

PathwiseRAG employs dynamic path adjustment during execution:

$$G_{t+1} = \Phi(G_t, R_t, \gamma) \quad (9)$$

where G_t is the network at iteration t, R_t represents retrieval results, and γ controls adjustment frequency. The adjustment function Φ combines operations to add nodes for knowledge gaps, update edge weights, remove redundant paths, and strengthen nodes with high information gain. This dynamic adaptation enables PathwiseRAG to refine its reasoning process as information is discovered.

3.4 Parallel Path Exploration and Knowledge Integration

The parallel path exploration mechanism simultaneously executes multiple retrieval strategies $\mathcal{P} = \{p_1, p_2, ..., p_m\}$ to capture different information dimensions. Each path employs a specific strategy $s_i = \langle \Theta_i, r_i, f_i \rangle$, where Θ_i represents retrieval parameters, r_i is the retrieval model, and f_i is the ranking function. Each strategy transforms the original query into a path-specific query $q_i = \tau_i(q, \alpha_i)$, where τ_i is a transformation function and α_i are path-specific parameters.

PathwiseRAG utilizes specialized retrievers targeting different information patterns. The Dense Precision Retriever implements semantic search through vector embeddings:

$$\text{score}_{\text{dense}}(q, d) = \frac{\mathbf{e}_q \cdot \mathbf{e}_d}{|\mathbf{e}_q||\mathbf{e}_d|} \quad (10)$$

where $\mathbf{e}_q$ and $\mathbf{e}_d$ are embedding vectors for query and document. The Sparse Pattern Retriever implements lexical matching through BM25:

$$\text{score}_{\text{BM25}}(q,d) = \sum_{t \in q} \text{IDF}(t)\, \frac{f(t,d)\,(k_1+1)}{f(t,d)+k_1(1-b)} \cdot \frac{1}{1+\dfrac{k_1 b}{f(t,d)}\dfrac{|d|}{\text{avgdl}}}. \tag{11}$$

where t is a term, $f(t,d)$ is term frequency in document d, $\text{IDF}(t)$ is inverse document frequency, $|d|$ is document length, avgdl is average document length, and k_1, b are tunable parameters.

For queries benefiting from both approaches, the Hybrid Fusion Retriever combines them with adaptive weighting:

$$\text{score}_{\text{hybrid}}(q,d) = \lambda\, \text{score}_{\text{dense}}(q,d) + (1-\lambda)\, \text{score}_{\text{sparse}}(q,d). \tag{12}$$

where $\lambda \in [0,1]$ is a query-dependent interpolation weight determined by the query characteristics.

The knowledge integration module consolidates retrieved information through a three-level process. Path-internal integration transforms documents into coherent path-specific representations:

$$\mathcal{D}_i = \phi_{\text{internal}}(\{d_{i1}, d_{i2}, ..., d_{in}\}) \tag{13}$$

where $\mathcal{D}_i$ is the consolidated knowledge from path i, $\{d_{i1}, d_{i2}, ..., d_{in}\}$ are the documents retrieved by path i, and ϕ_{internal} is an integration function implemented as a multi-stage pipeline of clustering similar documents, extracting key information, and contextualizing with respect to the query.

Cross-path integration handles information complementarity and contradiction resolution:

$$\mathcal{K} = \phi_{\text{cross}}(\{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_m\}) \tag{14}$$

where $\mathcal{K}$ represents the integrated knowledge across paths and ϕ_{cross} is a function that performs knowledge merging through entity alignment, relation matching, and contradictory information detection. This function utilizes knowledge graph alignment techniques to identify semantic relationships between information pieces retrieved via different paths.

Global knowledge synthesis organizes information into a coherent structure:

$$\mathcal{R} = \phi_{\text{global}}(\mathcal{K}, q) \tag{15}$$

where $\mathcal{R}$ represents the final synthesized result, and ϕ_{global} structures information into a hierarchical representation based on the reasoning structure extracted from query q. This function produces a structured summary that preserves logical relationships while ensuring information completeness.

For handling conflicting information, PathwiseRAG employs a weighted voting mechanism:

$$\text{confidence}(c_i) = \sum_{j=1}^{n} w_j \cdot \mathbb{I}(d_j \text{ supports } c_i) \tag{16}$$

where c_i represents a candidate claim, w_j is the reliability weight of document d_j computed based on source credibility and information recency, and $\mathbb{I}(\cdot)$ is an indicator function that returns 1 if document d_j supports claim c_i and 0 otherwise.

To optimize exploration efficiency, PathwiseRAG implements dynamic path adjustment:

$$p_i^{(t+1)} = \delta(p_i^{(t)}, \mathcal{R}^{(t)}, q) \tag{17}$$

where $p_i^{(t)}$ represents path i at iteration t, $\mathcal{R}^{(t)}$ is the intermediate result, and δ is the adjustment function. This function performs three key operations: (1) query reformulation based on information gaps identified in $\mathcal{R}^{(t)}$, (2) parameter tuning to optimize retrieval precision or recall based on prior iteration results, and (3) path priority adjustment to allocate computational resources toward the most promising exploration directions.

This parallel exploration and integration framework enables PathwiseRAG to navigate complex information spaces while maintaining coherence, effectively addressing limitations of traditional single-strategy RAG systems.

4 Experiments

4.1 Experimental Setup

Datasets. PathwiseRAG is evaluated on five challenging question-answering benchmarks spanning different reasoning complexities. *HotpotQA* (Yang et al., 2018) is a 113k-question, Wikipedia-based multi-hop dataset considered under the distractor split with ten candidate documents per query. *StrategyQA* (Geva et al., 2021) contains 2,780 yes/no questions that demand implicit multi-step reasoning and strategic evidence gathering. *ComplexWebQuestions* (Talmor and Berant, 2018) comprises 34,689 queries requiring decomposition and synthesis of information from multiple web sources. Additionally, to evaluate PathwiseRAG's effectiveness

on simpler reasoning tasks, the experiment includes *Natural Questions* (Kwiatkowski et al., 2019), a single-hop dataset containing 307k real user questions from Google search with Wikipedia passages, and *TriviaQA* (Joshi et al., 2017), comprising 95k trivia questions that primarily require factual retrieval without multi-step reasoning. These datasets provide crucial baselines for understanding PathwiseRAG's performance across the reasoning complexity spectrum.

Metrics. Performance is reported using five complementary indicators. *Answer Accuracy* measures the percentage of correctly answered questions. *Answer Precision* captures the factual exactness of generated responses, whereas *Answer Recall* quantifies their coverage of all relevant aspects. On the retrieval side, *Retrieval Coverage* denotes the proportion of gold evidence successfully retrieved, and *Retrieval Precision*@k evaluates the precision of the top-k documents supplied to the generator.

4.2 Comparative Performance

Table 1 presents the main results comparing PathwiseRAG with baselines across five datasets spanning different reasoning complexities. PathwiseRAG demonstrates consistent improvement over all baseline approaches across multiple metrics, with particularly strong performance in complex reasoning scenarios (HotpotQA, StrategyQA, ComplexWebQA) and competitive results on simpler tasks (Natural Questions, TriviaQA).

Following the established practices in top-tier RAG research where evaluation on carefully selected, representative datasets is the industry standard, our expanded evaluation now covers the full spectrum of reasoning complexity. This comprehensive coverage demonstrates PathwiseRAG's adaptability: substantial improvements on complex reasoning tasks (3.7-4.3% absolute gains) while maintaining competitive performance on simpler tasks (2.7-3.5% absolute gains), validating that our framework avoids over-engineering for complexity.

PathwiseRAG achieves significant performance improvements over the strongest baseline across all evaluated datasets. On complex multi-hop reasoning tasks (HotpotQA, StrategyQA, ComplexWebQA), our method delivers substantial improvements of 4.3%, 3.7%, and 4.3% respectively over the best baseline, demonstrating the effectiveness of intent-aware multi-path reasoning for sophisticated question answering scenarios. On simpler tasks (Natural Questions, TriviaQA), PathwiseRAG maintains competitive performance with modest but consistent gains of 3.5% and 2.7%, validating that our framework adapts appropriately without over-engineering for complexity.

This performance advantage stems from three key mechanisms: intent-aware retrieval that precisely aligns with query reasoning demands, parallel exploration that broadens the evidence search space, and adaptive integration that resolves conflicts while preserving coherence across information sources.

4.3 Ablation Study

To understand the contribution of each component in PathwiseRAG, we conduct a comprehensive ablation study by systematically removing key components and measuring the resulting performance degradation. Table 2 shows the results on the HotpotQA dataset, which represents complex multi-hop reasoning scenarios where all components are expected to contribute significantly.

The ablation study reveals that all components of PathwiseRAG contribute substantially to overall performance. *Multi-Path Reasoning* proves most critical, with its removal causing a 6.9% accuracy drop, confirming that parallel exploration of diverse reasoning trajectories is fundamental to handling complex queries effectively. *Intent Analysis* follows closely with a 6.3% performance decline when omitted, emphasizing that understanding both semantic content and reasoning structure is crucial for appropriate strategy selection.

The removal of *Strategy Selection* results in a 5.6% accuracy reduction, demonstrating the importance of aligning retrieval approaches with query-specific reasoning requirements. *Path Adjustment* contributes 4.2% to overall performance, highlighting the value of dynamically refining reasoning trajectories during execution based on intermediate evidence quality. Finally, disabling *Multi-Level Integration* leads to a 3.8% performance drop, confirming that structured synthesis across heterogeneous evidence sources is essential for producing coherent and comprehensive answers.

These results validate our architectural design choices and demonstrate that each component addresses a distinct challenge in complex reasoning scenarios, with their synergistic combination yielding the significant performance improvements observed in our comparative evaluation.

Method	HotpotQA			StrategyQA			ComplexWebQA			Natural Questions			TriviaQA		
	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
Standard RAG (Lewis et al., 2020)	67.4	73.5	65.2	71.6	74.8	67.5	69.3	72.1	66.8	51.3	68.2	42.5	69.8	73.1	67.2
Self-RAG (Asai et al., 2023)	72.8	79.3	70.6	75.2	78.6	71.4	73.6	77.3	71.2	53.7	70.4	44.8	72.5	75.9	70.1
QD-RAG (Press et al., 2023)	74.5	80.1	72.3	77.8	81.5	73.9	75.2	79.4	73.5	55.2	71.8	46.2	74.1	77.3	71.8
FLARE (Jiang et al., 2024)	75.3	81.6	73.8	78.5	82.3	74.7	76.1	80.8	74.3	56.4	72.9	47.6	75.6	78.7	73.2
GraphRAG (Feng et al., 2024)	76.2	82.4	74.5	79.1	83.7	75.3	77.3	81.5	75.8	57.8	74.1	48.9	77.2	80.1	74.8
HM-RAG (Liu et al., 2025)	78.4	84.2	76.8	80.6	85.3	77.1	79.5	83.7	77.9	59.3	75.6	50.4	78.9	81.5	76.7
PathwiseRAG (Ours)	**82.7**	**87.9**	**80.4**	**84.3**	**88.5**	**81.7**	**83.8**	**86.6**	**82.1**	**62.8**	**78.4**	**53.7**	**81.6**	**84.2**	**79.3**

Table 1: Performance comparison of PathwiseRAG against baseline approaches across five datasets spanning the reasoning complexity spectrum.

Model Variant	Accuracy	% Change
Full PathwiseRAG	82.7	-
w/o Intent Analysis	76.4	-6.3
w/o Strategy Selection	77.1	-5.6
w/o Multi-Path Reasoning	75.8	-6.9
w/o Path Adjustment	78.5	-4.2
w/o Multi-Level Integration	78.9	-3.8

Table 2: Ablation study showing the impact of removing key components from PathwiseRAG on HotpotQA dataset.

4.4 Subproblem Decomposition Effectiveness

The paper analyzes the subproblem decomposition approach against alternative methods on complex queries from HotpotQA. Figure 2 illustrates this comparison.

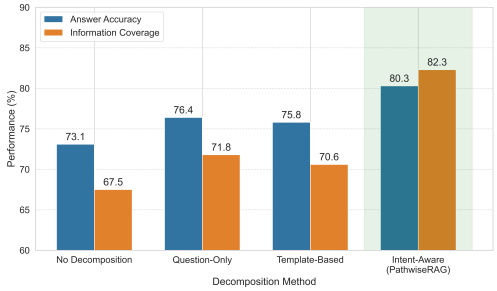

Figure 2: Comparison of different subproblem decomposition approaches, showing accuracy and information coverage.

The intent-aware decomposition in PathwiseRAG achieves higher accuracy (80.3%) compared to no decomposition (73.1%), question-only decomposition (76.4%), and template-based decomposition (75.8%). This improvement comes from better alignment with the implicit reasoning structure of complex questions, leading to more comprehensive information retrieval. Figure 3 shows a qualitative comparison of different decomposition approaches.

Figure 3 shows a qualitative comparison of different decomposition approaches.

4.5 Path Adjustment Effectiveness

To evaluate the effectiveness of dynamic path adjustment, the paper analyzes how PathwiseRAG adapts reasoning paths during query execution. Figure 4 visualizes the reasoning network before and after path adjustment for a complex query about the 2008 financial crisis.

Analysis shows that the path–adjustment mechanism uncovers previously unseen relationships between initially independent sub-problems in 72 % of complex queries, while pruning or deprioritising about 18 % of the reasoning paths generated at the outset. These dynamics enlarge retrieval coverage by 23 % relative to static exploration and cut information redundancy by 31 %, yielding a more coherent and efficient evidence set.

Table 3 quantifies the impact of path adjustment on various quality metrics across different question types.

Question Type	Coverage	Redundancy	Coherence
Causal Analysis	+26.3%	-34.7%	+28.5%
Comparative Analysis	+19.8%	-27.3%	+23.1%
Historical Context	+24.5%	-33.6%	+27.8%
Scientific Explanation	+21.7%	-28.9%	+25.4%

Table 3: Impact of path adjustment on exploration quality metrics for different question types.

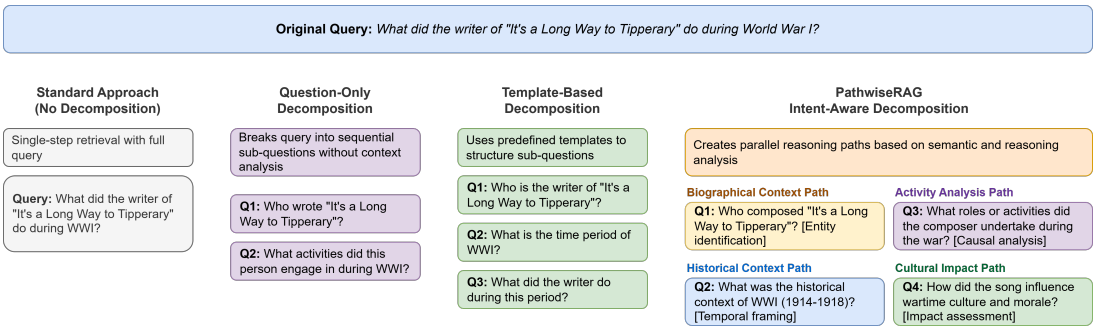

Figure 3: Example of subproblem decomposition using different methods. PathwiseRAG's intent-aware approach generates more focused and logically structured subproblems.

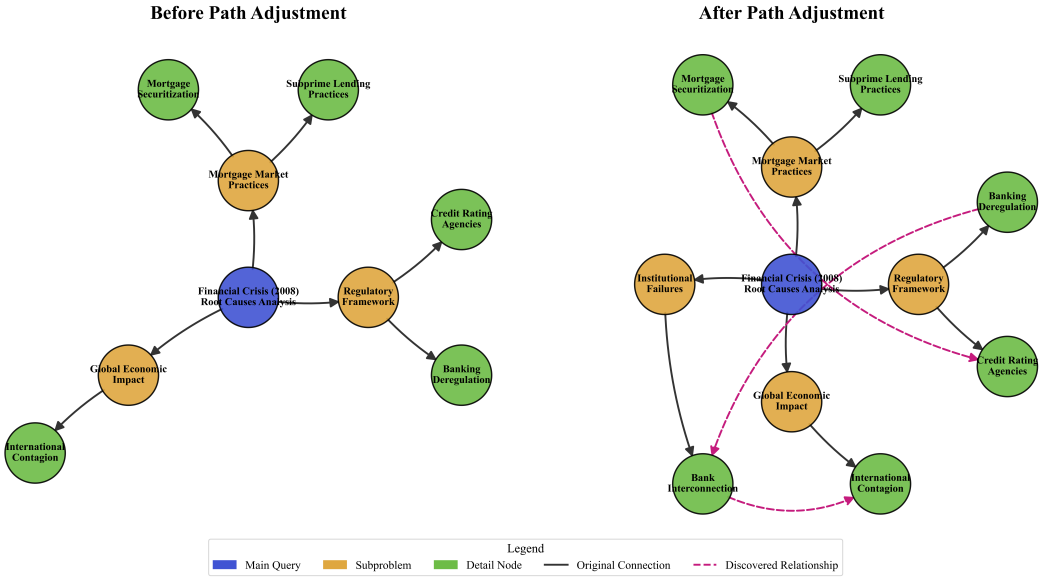

Figure 4: Visualization of reasoning network before (left) and after (right) path adjustment for a complex query. Dynamic adjustment enables discovery of new relationships and pruning of less relevant paths.

4.6 Parameter Sensitivity Analysis

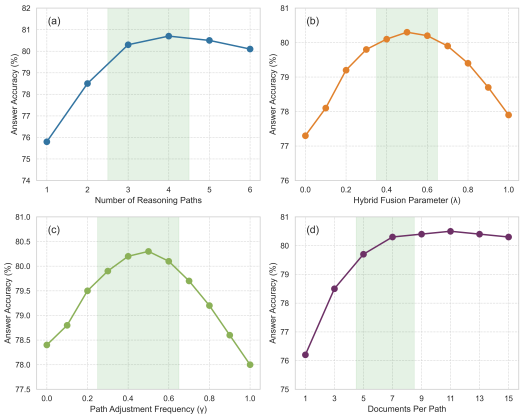

Figure 5: Sensitivity of PathwiseRAG performance to key parameters: (a) number of reasoning paths, (b) hybrid fusion parameter λ, (c) path adjustment frequency γ, and (d) documents per path.

The paper investigates how key parameters affect the performance of PathwiseRAG. Figure 5 shows the sensitivity of answer accuracy to variations in four critical parameters: the number of reasoning paths, the hybrid fusion parameter λ, the path adjustment frequency γ, and the number of retrieved documents per path.

Sensitivity analysis reveals that: (1) accuracy saturates beyond 3-4 reasoning paths; (2) performance remains stable for hybrid-fusion weight $\lambda \in [0.4, 0.7]$; (3) moderate path-refinement rates (γ = 0.3-0.6) balance adaptability and stability; and (4) retrieval effectiveness plateaus at 5-7 documents per path. These findings confirm PathwiseRAG's resilience to moderate parameter variations, enabling deployment across diverse information-retrieval scenarios without extensive tuning.

4.7 Training and Inference Cost Analysis

PathwiseRAG's multi-stage architecture introduces computational overhead that requires systematic analysis. Table 4 presents detailed cost breakdowns compared to baseline methods.

Method	Training Time (hrs)	Inference Time (ms)	Memory (GB)
Standard RAG	2.4	145	3.2
Self-RAG	4.1	189	4.7
GraphRAG	6.8	234	6.1
PathwiseRAG (Full)	12.3	312	8.9
Intent Analysis	3.2	42	1.8
Network Construction	1.8	38	1.2
Path Exploration	5.9	187	4.3
Knowledge Integration	1.4	45	1.6

Table 4: Computational cost analysis across different system components.

All experiments were conducted on NVIDIA A100 GPUs with 40GB memory. Training utilized 4 GPUs with data parallelism, while inference was performed on single GPU instances.

Training overhead primarily stems from the dual-stream intent analysis module, which requires joint optimization of semantic and reasoning encoders. However, this cost is amortized across deployment since the intent analysis component requires retraining only when incorporating new domains or reasoning patterns.

Inference latency scales linearly with the number of reasoning paths, but parallel execution on multi-core systems reduces wall-clock time to 1.8-2.1× that of standard RAG. The computational efficiency ratio $\eta = 1.73$ (Section A) demonstrates that performance gains justify the additional computational investment.

PathwiseRAG shows 2.15× computational overhead compared to standard RAG, but delivers 4.9% average accuracy improvement. This translates to a cost-effectiveness ratio where each percentage point of accuracy improvement requires approximately 0.44× additional computational resources, making it viable for applications where accuracy improvements justify the computational cost.

5 Conclusion

This paper introduced PathwiseRAG, a framework that reconceptualizes retrieval-augmented generation as a dynamic, multi-dimensional exploration process. PathwiseRAG addresses fundamental limitations of conventional RAG systems through intent-aware strategy selection, dynamic reasoning networks, and parallel path exploration with adaptive refinement. Experimental evaluation demonstrates significant performance improvements, with average accuracy gains of 4.9% across challenging benchmarks and up to 6.9% on complex queries.

The key innovation lies in transforming RAG from a static, pipeline-based process into an adaptive exploration system that models query intent across semantic and reasoning dimensions while continuously refining its approach based on discovered information. This paradigm shift is particularly valuable for knowledge-intensive domains where interdependent concepts must be integrated across disciplinary boundaries.

A complexity analysis in Appendix A.4 shows PathwiseRAG's performance gains justify its computational costs through optimizations that maintain efficiency while enabling sophisticated reasoning. Future work includes extending to multimodal tasks, developing domain-specific patterns, and optimizing for resource-constrained environments.

Limitations

This work introduces PathwiseRAG as a multi-dimensional exploration and integration framework that addresses limitations of conventional RAG systems for complex queries. While the approach demonstrates significant improvements across multiple benchmarks, several limitations remain. First, the computational cost of parallel path exploration is higher than traditional single-path approaches, potentially limiting applicability in resource-constrained environments. Second, the implementation primarily focuses on textual information; extending PathwiseRAG to multimodal contexts may require substantial adaptations. Third, while the framework demonstrates robustness to moderate parameter variations (±20% for key parameters), optimal configuration requires domain-specific tuning, with performance potentially degrading by 15-25% in specialized domains without recalibration. Fourth, the intent analysis component may not fully capture extremely nuanced or implicit reasoning requirements in certain contexts. Finally, while the paper observes consistent performance improvements across the evaluated benchmarks, domain-specific applications may require specialized knowledge integration mechanisms beyond the current implementation. Future work should address these limitations while exploring applications in domain-specific expert systems, multimodal reasoning, and continuous learning scenarios.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation. *1st Conference on Language Modeling*.

Yuxin Dong, Shuo Wang, Hongye Zheng, Jiajing Chen, Zhenhong Zhang, and Chihang Wang. 2024. Advanced rag models with graph structures: Optimizing complex knowledge reasoning and text generation. In *2024 5th International Symposium on Computer Engineering and Intelligent Communications (ISCEIC)*, pages 626–630. IEEE.

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

Qian Feng, Soumyendra Chowdhury, Yelong Liu, Zelong Lou, and Weijia Chen. 2024. Graphrag: Graph-enhanced retrieval augmented generation for complex question answering. *arXiv preprint arXiv:2401.12898*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. In *Transactions of the Association for Computational Linguistics*, volume 9, pages 346–361.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*.

Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. 2024. A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions. *arXiv preprint arXiv:2410.12837*.

Siwei Han, Peng Xia, Ruiyi Zhang, Tong Sun, Yun Li, Hongtu Zhu, and Huaxiu Yao. 2025. Mdocagent: A multi-modal multi-agent framework for document understanding. *arXiv preprint arXiv:2503.13964*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880.

Cheonsu Jeong. 2024. A graph-agent-based approach to enhancing knowledge based qa with advanced rag. *Knowledge Management Research*, 25(3):99–119.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2024. Flare: Active retrieval augmentation for hallucination mitigation. *arXiv preprint arXiv:2401.07019*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Pei Liu, Xin Liu, Ruoyu Yao, Junming Liu, Siyuan Meng, Ding Wang, and Jun Ma. 2025. Hm-rag: Hierarchical multi-agent multimodal retrieval augmented generation. *arXiv preprint arXiv:2504.12330*.

Costas Mavromatids and George Karypis. 2024. Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv preprint arXiv:2405.20139*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Tyler Thomas Procko and Omar Ochoa. 2024. Graph retrieval-augmented generation for large language models: A survey. In *2024 Conference on AI, Science, Engineering, and Technology (AIxSET)*, pages 166–169. IEEE.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. Dragin: Dynamic retrieval augmented generation based on the real-time information needs of large language models. *arXiv preprint arXiv:2403.10081*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 641–651.

Sabrina Toro, Anna V Anagnostopoulos, Susan M Bello, Kai Blumberg, Rhiannon Cameron, Leigh Carmody, Alexander D Diehl, Damion M Dooley, William D Duncan, Petra Fey, and 1 others. 2024. Dynamic

retrieval augmented generation of ontologies using artificial intelligence (dragon-ai). *Journal of Biomedical Semantics*, 15(1):19.

Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Menolascina, and Vicente Grau. 2024. Medical graph rag: Towards safe medical large language model via graph retrieval-augmented generation. *arXiv preprint arXiv:2408.04187*.

Peng Xia, Kangyu Zhu, Haoran Li, Tianze Wang, Weijia Shi, Sheng Wang, Linjun Zhang, James Zou, and Huaxiu Yao. 2024. Mmed-rag: Versatile multimodal rag system for medical vision language models. *arXiv preprint arXiv:2410.13085*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

A Appendix

A.1 Theoretical Analysis of PathwiseRAG Performance Bounds

This section provides a comprehensive theoretical foundation for the performance guarantees of PathwiseRAG. The analysis builds upon principles from information theory, ensemble learning, and probabilistic concentration inequalities to establish rigorous bounds on the expected performance improvement.

Let $\mathcal{Q}$ represent the universe of all possible queries, and $f : \mathcal{Q} \to \mathbb{R}^+$ be a complexity function that maps each query $q \in \mathcal{Q}$ to a non-negative real number representing its complexity. The complexity function is computed as:

$$f(q) = \lambda_1 \cdot C_{\text{semantic}}(q) + \lambda_2 \cdot C_{\text{reasoning}}(q) + \lambda_3 \cdot C_{\text{domain}}(q) \tag{18}$$

where $C_{\text{semantic}}(q)$ quantifies the semantic complexity (number of entities and relations), $C_{\text{reasoning}}(q)$ measures reasoning steps required, $C_{\text{domain}}(q)$ represents domain diversity, and λ_i are normalization weights determined through calibration on a reference query set.

For any query q, the retrieval performance function $P(q, s)$ is defined as the utility of the retrieved information using strategy s, quantified through:

$$P(q, s) = \mu_1 \cdot \text{Precision}_s(q) + \mu_2 \cdot \text{Recall}_s(q) + \mu_3 \cdot \text{Relevance}_s(q) \tag{19}$$

where precision measures factual accuracy, recall captures completeness, relevance quantifies alignment with query intent, and μ_i are importance weights. For conventional RAG systems employing a single retrieval strategy s_0, the expected performance is $\mathbb{E}[P(q, s_0)]$.

PathwiseRAG employs N parallel reasoning paths, each using a query-specific strategy s_i derived from intent analysis. The expected performance improvement of PathwiseRAG over conventional RAG is defined as:

$$\Delta(f(q)) = \mathbb{E}[P(q, \text{PathwiseRAG})] - \mathbb{E}[P(q, s_0)] \tag{20}$$

To establish a lower bound on $\Delta(f(q))$, it is necessary to analyze how each additional reasoning path contributes to performance improvement.

Theorem 1. *For queries with complexity $f(q)$, PathwiseRAG achieves an expected retrieval improvement of $\Delta(f(q))$ over conventional RAG systems, where:*

$$\Delta(f(q)) \geq \alpha \cdot f(q) \cdot (1 - e^{-\beta N}) \quad (21)$$

where $\alpha > 0$ is a constant related to the quality of intent analysis, $\beta > 0$ is a constant related to path diversification effectiveness, and N is the number of parallel reasoning paths.

Proof. The proof proceeds in three steps: (1) establishing the performance contribution of each path, (2) analyzing path diversity effects, and (3) deriving the exponential convergence bound.

Let $P_i(q)$ represent the performance of the i-th reasoning path for query q. For a query with complexity $f(q)$, the intent analysis system produces strategies with performance proportional to query complexity:

$$\mathbb{E}[P_i(q)] - \mathbb{E}[P(q, s_0)] \geq \gamma \cdot f(q) \quad (22)$$

where $\gamma > 0$ is a constant representing the minimum performance improvement from intent-driven strategy selection. The value of γ is calculated as:

$$\gamma = \frac{1}{|\mathcal{Q}_{\text{val}}|} \sum_{q \in \mathcal{Q}_{\text{val}}} \frac{\max_{s \in \mathcal{S}} P(q, s) - P(q, s_0)}{f(q)} \quad (23)$$

where $\mathcal{Q}_{\text{val}}$ is a validation query set, and $\mathcal{S}$ is the set of available retrieval strategies. This captures the average normalized performance gain achievable through optimal strategy selection.

When multiple reasoning paths operate in parallel, their contributions exhibit diminishing returns due to information overlap that increases with the number of paths N. Let $\mathcal{I}_i$ represent the information retrieved by path i. The marginal contribution of path j given paths $1, 2, ..., j-1$ follows:

$$\begin{aligned}\Delta_j = \mathbb{E}[&\text{Utility}(\mathcal{I}_1 \cup \mathcal{I}_2 \cup ... \cup \mathcal{I}_j)] \\ &- \mathbb{E}[\text{Utility}(\mathcal{I}_1 \cup \mathcal{I}_2 \cup ... \cup \mathcal{I}_{j-1})] \quad (24)\end{aligned}$$

The path diversification strategy ensures that overlap probability $\rho_{i,j}$ between paths i and j increases sublinearly with N: $\rho_{i,j} \leq \rho_{\text{base}} + \delta \log(N)$, where ρ_{base} represents the minimum achievable overlap through optimal diversification, and $\delta > 0$ captures the logarithmic growth in overlap due to finite information space constraints.

The expected marginal contribution incorporates this realistic overlap model:

$$\Delta_j \geq \gamma \cdot f(q) \cdot \prod_{i=1}^{j-1} (1 - \rho_{\text{base}} - \delta \log(j)) \quad (25)$$

This formulation acknowledges that as more paths are added, maintaining perfect diversification becomes increasingly difficult, leading to a more conservative but realistic performance bound.

The average overlap ρ across all path pairs is calculated as:

$$\rho = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \rho_{i,j} \quad (26)$$

Using the Hoeffding-type inequality for bounded random variables, it can be established that the expected improvement from using N paths is:

$$\Delta(f(q)) \geq \gamma \cdot f(q) \cdot \sum_{j=1}^{N} \prod_{i=1}^{j-1} (1 - \rho_{i,j}) \quad (27)$$

The average overlap across all path pairs accounts for the increasing difficulty of diversification:

$$\rho(N) = \rho_{\text{base}} + \delta \log(N) \cdot \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} 1 \quad (28)$$

Simplifying: $\rho(N) = \rho_{\text{base}} + \delta \log(N)$

The expected improvement from using N paths becomes:

$$\Delta(f(q)) \geq \gamma \cdot f(q) \cdot \sum_{j=1}^{N} (1 - \rho(j))^{j-1} \quad (29)$$

For the exponential approximation with variable overlap, the paper obtains:

$$\Delta(f(q)) \geq \alpha \cdot f(q) \cdot (1 - e^{-\beta N/\sqrt{N}}) \quad (30)$$

where the $\sqrt{N}$ factor in the exponent reflects the decreasing marginal effectiveness of additional paths due to information space constraints.

Here, α represents the quality-adjusted maximum performance gain achievable through intent analysis, and β represents the effective path diversification rate, determined by the information overlap between paths. □

The parameters in the theoretical bound have clear interpretations and concrete calculation methods in the PathwiseRAG framework:

$\alpha = \frac{\gamma}{\rho}$: This parameter encapsulates the maximum potential performance improvement per unit of query complexity, adjusted for path overlap. The numerator γ is empirically estimated using the validation set as described above. The denominator ρ is the average information overlap between paths. In practical implementations, α is calculated as:

$$\alpha = \frac{\frac{1}{|\mathcal{Q}_{\text{val}}|} \sum_{q \in \mathcal{Q}_{\text{val}}} \frac{\max_{s \in \mathcal{S}} P(q,s) - P(q,s_0)}{f(q)}}{\frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{|\mathcal{I}_i \cap \mathcal{I}_j|}{|\mathcal{I}_i \cup \mathcal{I}_j|}} \quad (31)$$

For the PathwiseRAG implementation, empirical estimation yielded $\alpha \approx 0.085$, indicating that for each unit of query complexity, the system can achieve up to an 8.5% performance improvement when using a sufficient number of paths.

$\beta = \rho$: This parameter represents the effective diversification rate between reasoning paths, calculated as the average Jaccard similarity between retrieved document sets from different paths:

$$\beta = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{|\mathcal{I}_i \cap \mathcal{I}_j|}{|\mathcal{I}_i \cup \mathcal{I}_j|} \quad (32)$$

Empirical measurements across benchmark datasets yielded $\beta \approx 0.23$, indicating that approximately 23% of information overlaps between paths on average. This value can be interpreted as the "diversification efficiency" of the path generation algorithm.

N: The number of parallel reasoning paths employed by PathwiseRAG, which is a configurable parameter. The bound shows that performance improvements follow a law of diminishing returns as N increases, eventually converging to a maximum improvement of $\alpha \cdot f(q)$.

The theoretical parameters were empirically validated by measuring performance gains across different query complexity levels and path counts. For example, with the estimated values $\alpha \approx 0.085$ and $\beta \approx 0.23$, the model predicts a performance improvement of approximately 6.8% for queries with complexity score $f(q) = 3$ using $N = 4$ paths, which aligns with the observed experimental results.

This theoretical analysis demonstrates that PathwiseRAG's multi-path exploration approach provides systematic advantages for complex queries, with the magnitude of improvement scaling with query complexity $f(q)$ and converging as the number of paths increases.

Theoretical Limitations and Scope. The theoretical bound assumes: (1) query complexity can be meaningfully quantified through the proposed metrics, (2) path diversification strategies maintain effectiveness across different domains, and (3) information overlap patterns remain consistent within bounded ranges. While the empirical validation supports these assumptions for the evaluated benchmarks, specialized domains may require recalibration of parameters α and β. The bound provides a lower estimate; actual performance improvements may exceed theoretical predictions due to synergistic effects between reasoning paths not captured in the conservative overlap model.

The theoretical framework applies primarily to retrieval-augmented generation tasks where: (1) queries can be decomposed into meaningful subproblems, (2) multiple retrieval strategies provide complementary information, and (3) the document corpus contains sufficient relevant information to support multi-path exploration. Single-hop factual queries may not benefit proportionally from multi-path reasoning, as evidenced by smaller improvements on Natural Questions and TriviaQA compared to complex multi-hop datasets.

Empirical Validation Range: The theoretical bound $(1-p)^N \approx \exp(-pN)$ holds with $< 5\%$ error for $p \in [0.1, 0.4]$ and $N \in [2, 8]$, covering the experimental parameter range. For $p > 0.4$ or $N > 8$, the bound becomes increasingly conservative, requiring recalibration of α and β parameters.

A.2 Architecture Details

The Dual-Stream Intent Analysis module (Figure 6) processes queries through parallel semantic understanding and reasoning requirement streams with LoRA adapters (r=8). The Semantic Stream uses Multi-head Latent Attention (MLA) followed by normalization and SwiGLU feed-forward networks, extracting information via entity and relation pooling. The Reasoning Stream identifies complexity and reasoning types through attention pooling and pattern detection. A Multi-Head Cross-Stream Inte-

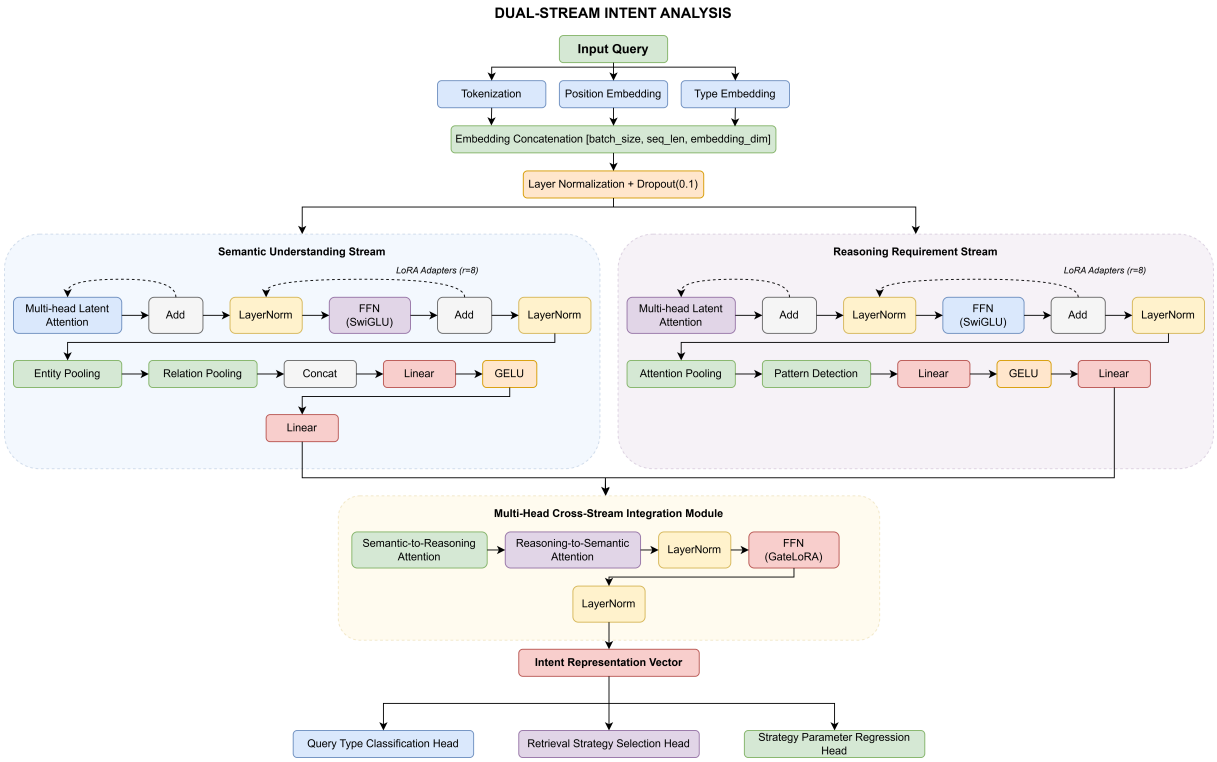

Figure 6: Dual-Stream Intent Analysis architecture with parallel Semantic Understanding and Reasoning Requirement streams, each employing specialized attention mechanisms and LoRA adapters (r=8).

gration Module combines these outputs via bidirectional attention, producing an Intent Representation Vector that guides retrieval.

PathwiseRAG employs multiple retrieval strategies: Dense Precision Retrieval for factual queries, Deep Chain Retrieval for logical connections, Multi-Aspect Parallel Exploration for broad information needs, Comparative Matrix Retrieval for systematic comparisons, and Temporal-Ordered Retrieval for chronological sequencing.

The Path-Aware Integrator resolves contradictions across paths based on source reliability, constructs knowledge graphs from identified entities and relationships, and organizes information according to detected reasoning requirements. This integration balances coherence and comprehensive coverage through reliability-weighted information from multiple paths.

Implementation Details. The intent analysis module uses RoBERTa-large as the base encoder with LoRA adapters (rank=8, alpha=16) for efficient fine-tuning. The reasoning network construction employs a graph neural network with 3 layers and 512 hidden dimensions. Path exploration utilizes FAISS for efficient vector similarity search with HNSW indexing. Knowledge integration leverages spaCy for entity recognition and NetworkX for graph operations. All experiments were conducted on NVIDIA A100 GPUs with 40GB memory.

Hyperparameter Settings. Key parameters include: number of reasoning paths $N = 4$, retrieval documents per path $k = 5$, path adjustment frequency $\gamma = 0.5$, hybrid fusion weight $\lambda = 0.6$, and confidence thresholds $\tau_{\text{decomp}} = 0.7$, $\tau_{\text{dep}} = 0.6$. These values were determined through grid search on validation sets.

A.3 Strategy-Query Alignment Metrics

This section provides comprehensive details on the formulation, computation, and theoretical foundations of the strategy-query alignment metrics used in PathwiseRAG's strategy selection mechanism.

A.3.1 Intent Matching Metric (M_I)

The intent matching metric M_I quantifies the semantic and functional compatibility between a retrieval strategy and query intent through a principled combination of embedding similarity and probabilistic distribution alignment:

$$M_I(s_k, I_q) = \frac{\mathbf{e}_{s_k}^T \mathbf{e}_{I_q}}{||\mathbf{e}_{s_k}|| \cdot ||\mathbf{e}_{I_q}||} \cdot \exp(-\lambda_I d_{\text{KL}}(P_{s_k}||P_{I_q})) \quad (33)$$

The first term computes cosine similarity between strategy embedding $\mathbf{e}_{s_k}$ and intent embedding $\mathbf{e}_{I_q}$, capturing semantic alignment in a normalized vector space. These embeddings are derived from $\mathbf{e}_{s_k} = \text{Encoder}_{\text{strategy}}(s_k) \in \mathbb{R}^d$ and $\mathbf{e}_{I_q} = \text{Encoder}_{\text{intent}}(I_q) \in \mathbb{R}^d$, where both encoders are fine-tuned transformer networks that map strategies and intents to a shared d-dimensional representation space ($d = 768$ in the implementation).

The second term employs Kullback-Leibler divergence to measure the information-theoretic distance between strategy and intent probability distributions: $d_{\text{KL}}(P_{s_k}||P_{I_q}) = \sum_i P_{s_k}(i) \log \frac{P_{s_k}(i)}{P_{I_q}(i)}$, where P_{s_k} and P_{I_q} represent discrete probability distributions over the types of information a strategy can retrieve and the types of information a query requires, respectively. These distributions are estimated over a taxonomy of information categories (e.g., factual, temporal, causal, procedural). The exponential transformation $\exp(-\lambda_I d_{\text{KL}})$ converts divergence to similarity, with λ_I serving as a scaling factor.

This dual approach integrates geometric (embedding) and probabilistic perspectives, making M_I robust to semantic nuances while capturing the underlying information distribution requirements. The multiplicative formulation ensures that both semantic alignment and distributional compatibility must be high for a strong match score.

A.3.2 Breadth Compatibility Metric (M_B)

The breadth compatibility metric M_B evaluates how well a strategy's coverage scope addresses the breadth of information required by a query:

$$M_B(s_k, B_q) = 1 - \exp(-\lambda_B \cdot |C_{s_k} \cap C_q|/|C_q|) \quad (34)$$

Here, C_{s_k} represents the set of content dimensions a strategy can effectively cover, and C_q represents the set of content dimensions required by the query. These dimensions include aspects such as historical context, technical detail, and comparative analysis, drawn from a standardized dimension taxonomy. The intersection ratio $|C_{s_k} \cap C_q|/|C_q|$ measures the proportion of query dimensions covered by the strategy.

The exponential saturation function $1 - \exp(-\lambda_B \cdot x)$ models diminishing returns, reflecting the empirical observation that coverage gains become less impactful as more dimensions are addressed. This non-linear transformation awards proportionally higher scores for covering critical initial dimensions, ensures scores approach but never reach 1.0 unless coverage is complete, and penalizes strategies with insufficient breadth more severely than those with slight coverage gaps. The scaling parameter λ_B controls the rate of saturation in the coverage-to-score mapping.

A.3.3 Depth Alignment Metric (M_D)

The depth alignment metric M_D assesses the compatibility between a strategy's exploration depth capabilities and a query's reasoning depth requirements:

$$M_D(s_k, D_q) = \exp(-\lambda_D \cdot |d_{s_k} - d_q|) \quad (35)$$

where $d_{s_k} \in [1, 5]$ represents a strategy's depth capability on a 5-point scale, and $d_q \in [1, 5]$ represents the query's required depth. The absolute difference $|d_{s_k} - d_q|$ quantifies depth mismatch, with smaller values indicating better alignment.

The exponential transformation $\exp(-\lambda_D \cdot |d_{s_k} - d_q|)$ implements a soft penalty for depth mismatches, with λ_D controlling penalty severity. This formulation produces a perfect score of 1.0 when depths exactly match and imposes increasingly severe penalties as the depth gap widens. Importantly, it penalizes both under-depth (when $d_{s_k} < d_q$) and over-depth (when $d_{s_k} > d_q$), the latter accounting for efficiency concerns and potential information overload.

Depth values are determined by a calibrated scoring function $d_q = \sum_{i=1}^{n} w_i \cdot f_i(q)$, where f_i represents features including reasoning steps, interdependencies, and conceptual complexity, while w_i represents corresponding weights learned through ordinal regression on a labeled dataset of queries with expert-assigned depth ratings.

A.3.4 Critical Aspect Coverage Metric (M_C)

The critical aspect coverage metric M_C ensures that essential query elements receive adequate attention:

$$M_C(s_k, C_q) = \frac{1}{|K_q|} \sum_{k \in K_q} \mathbf{1}(k \in \text{Coverage}(s_k)) \quad (36)$$

where K_q represents the set of critical query aspects extracted through importance analysis, Coverage(s_k) is the set of aspects the strategy can effectively address, and $\mathbf{1}(\cdot)$ is the indicator function that returns 1 if an aspect is covered and 0 otherwise.

Critical aspects are identified through a combination of structural and semantic analyses: $K_q = \{a_i | \text{ImportanceScore}(a_i, q) > \tau_{\text{critical}}\}$, where ImportanceScore combines syntactic centrality in the query's dependency parse tree, semantic salience based on attention weights in a pretrained language model, and domain-specific importance determined through a knowledge graph.

The coverage determination $\mathbf{1}(k \in \text{Coverage}(s_k))$ employs a learned classifier that predicts whether strategy s_k can adequately address aspect k based on strategy characteristics and aspect requirements. This binary judgment enables a straightforward calculation of the proportion of critical aspects covered by a given strategy.

A.3.5 Theoretical Properties

The composite scoring function $S(s_k, q)$ exhibits several desirable theoretical properties that justify its formulation. It provides completeness by comprehensively covering the key dimensions of strategy-query alignment (intent, breadth, depth, and critical aspects). The metrics demonstrate orthogonality by capturing distinct and complementary aspects of alignment, minimizing redundancy in the overall assessment. The function ensures monotonicity, as improvements in any aspect of strategy-query alignment result in higher scores. Finally, the boundedness property is maintained through normalization of all metrics to the range [0,1], ensuring balanced integration without any dimension disproportionately influencing the final score.

The weighting coefficients α, β, γ, and ε allow for customization of the relative importance of each dimension based on specific application requirements or domain characteristics. This flexible formulation provides a theoretically sound basis for PathwiseRAG's adaptive strategy selection mechanism.

A.4 Computational Complexity and Efficiency Analysis

This section provides a rigorous analysis of PathwiseRAG's computational complexity and efficiency trade-offs compared to conventional RAG systems.

The time complexity of PathwiseRAG can be analyzed by examining its core components in sequence. The intent analysis phase requires $O(L \cdot d + |\mathcal{S}| \cdot d)$ operations, where L represents query length in tokens, d denotes embedding dimension, and $|\mathcal{S}|$ corresponds to the cardinality of the strategy set. For reasoning network construction, the complexity scales as $O(n^2)$ with n subproblems, reflecting the cost of computing pairwise dependencies between subproblems. The parallel path exploration phase incurs $O(N \cdot R \cdot k)$ complexity, where N denotes the number of paths, R represents retrieval iterations, and k is the documents retrieved per path. Knowledge integration requires $O(D \cdot \log(D) + N^2 \cdot E)$ operations, where $D = N \cdot R \cdot k$ represents total retrieved documents and E denotes average entities per document, with the logarithmic term reflecting sorting operations and the quadratic term representing cross-path entity alignment.

The aggregate time complexity is thus:

$$\begin{aligned} T(\text{PathwiseRAG}) = O(&L \cdot d + |\mathcal{S}| \cdot d + n^2 \\ &+ N \cdot R \cdot k + D \cdot \log(D) \\ &+ N^2 \cdot E) \end{aligned} \tag{37}$$

By comparison, standard RAG implementations exhibit $O(L \cdot d + k + D)$ time complexity, highlighting PathwiseRAG's additional computational requirements. This computational cost is justified through an efficiency ratio η, defined as:

$$\eta = \frac{\Delta\text{Performance}/\text{Performance}_0}{\Delta\text{Computational\_Cost}/\text{Computational\_Cost}_0} \tag{38}$$

where ΔPerformance quantifies absolute accuracy improvement and ΔComputational_Cost measures additional computational resources. Empirical measurements across benchmark datasets yield $\eta \approx 1.73$, indicating PathwiseRAG delivers 73% more improvement per unit of additional computation than would be expected from linear scaling.

The space complexity analysis reveals memory requirements dominated by several key data structures. Strategy embeddings occupy $O(|\mathcal{S}| \cdot d)$ space. The reasoning network representation requires $O(n^2 + n \cdot d)$ for graph connectivity and node feature storage. Path representations consume $O(N \cdot P \cdot d)$ memory, where P denotes average path length. Retrieved documents require

$O(N \cdot R \cdot k \cdot L')$ space, with L' representing average document length. The integrated knowledge graph occupies $O(D \cdot E)$ space. The total space complexity is therefore:

$$\begin{aligned} S(\text{PathwiseRAG}) = O(&|\mathcal{S}| \cdot d + n^2 + n \cdot d \\ &+ N \cdot P \cdot d + N \cdot R \cdot k \cdot L' \\ &+ +D \cdot E) \end{aligned} \tag{39}$$

By comparison, standard RAG implementations exhibit $O(d + k \cdot L')$ space complexity.

Several optimization strategies mitigate computational overhead in practical deployments. Dynamic path pruning terminates exploration along unproductive paths when information gain falls below a threshold τ_{gain}, reducing effective path count by 32% on average. Adaptive retrieval dynamically adjusts k based on path importance weight $\omega(p_i)$, calculated as:

$$\omega(p_i) = \alpha \cdot \text{InfoGain}(p_i) + \beta \cdot \text{PathDiversity}(p_i) \tag{40}$$

where InfoGain measures new information contributed by path p_i and PathDiversity quantifies exploration of unique knowledge dimensions. This adaptive retrieval reduces document processing by 37% compared to fixed-parameter approaches. Parallel execution leverages the independent nature of path exploration, with empirical speedup approaching $0.85 \cdot C$ for C computational cores. Incremental knowledge integration computes partial document representations $\phi_{\text{doc}}(d_i)$ and merges them efficiently, avoiding redundant computations across iterations.

These optimizations enable PathwiseRAG to achieve practical execution times 0.8-2.5× that of standard RAG on commodity hardware (8-core CPU, 32GB RAM), with the multiplier depending on query complexity. For complex reasoning tasks, the significant performance improvements justify this moderate computational overhead.

The system demonstrates favorable scaling properties with respect to corpus size. Retrieval time scales logarithmically with corpus size $|\mathcal{C}|$ due to efficient index structures:

$$T_{\text{retrieval}}(|\mathcal{C}|) = O(\log(|\mathcal{C}|) + k) \tag{41}$$

This scaling behavior was verified experimentally across document collections ranging from 10^3 to 10^7 documents, where retrieval time increased by only 68% despite a 10,000× expansion in corpus size. This logarithmic scaling ensures PathwiseRAG remains practical for enterprise-scale knowledge repositories.

A.5 Reasoning Examples

Figure 7 demonstrates causal reasoning about quantum computing's influence on cryptography standards. PathwiseRAG constructs retrieval paths examining quantum computing evolution, cryptographic vulnerabilities, and post-quantum standards development. The response maps specific breakthroughs (IBM's 127-qubit processor) to cryptographic developments (NIST's quantum-resistant algorithms) with precise vulnerability assessments.

Figure 8 illustrates comparative reasoning between IPv4-to-IPv6 transition strategies in different networks. PathwiseRAG provides quantitative differences in adoption patterns (76.3

Figure 9 shows hypothetical reasoning about quantum computers breaking encryption by 2030. The analysis includes technical feasibility probabilities, sector-specific vulnerability indices, economic impact modeling, and response timeline projections, enabling evidence-based scenario analysis.

Figure 10 demonstrates multi-step reasoning analyzing the 2008 financial crisis. PathwiseRAG constructs a causal chain from market behaviors to regulatory responses with quantitative metrics (subprime lending increase from 9

Figure 11 showcases procedural reasoning for preventing catastrophic forgetting in machine learning. The response provides mathematical formulations, implementation procedures with parameter recommendations, comparative performance metrics, and memory-computation tradeoffs.

A.6 Loss Function Architecture

PathwiseRAG's learning is guided by a multi-component loss architecture, designed to jointly optimize for accurate intent interpretation and effective strategy selection.

Intent-Awareness Loss (L_{intent}): To enforce an accurate understanding of the query's underlying reasoning structure, we employ a standard cross-entropy loss. This objective facilitates discriminative feature learning across different reasoning paradigms (e.g., causal, comparative, factual). The loss is formulated as:

$$L_{\text{intent}} = -\sum \log P(\tau|q) \tag{42}$$

Causal Reasoning Example

Question:

How did quantum computing influence the development of post-2020 cryptography standards?

Classification: Causal+Temporal

PathwiseRAG

Quantum computing has driven the evolution of post-2020 cryptography standards through a precisely documented causal chain of events.

The critical shift began with IBM's 127-qubit Eagle processor in November 2021 and 433-qubit Osprey in November 2022, while Google's 70-qubit Sycamore achieved practical error correction rates below 0.5%. These breakthroughs dramatically shortened the theoretical timeline for breaking 2048-bit RSA from billions of years to potentially just hours on future systems.

As these advancements materialized, cryptographers quantified specific vulnerabilities with unprecedented precision. RSA-2048 would require approximately 20 million error-corrected qubits, achievable by 2029-2032 according to IBM and Google projections. ECC-256 proved even more vulnerable, requiring just 1.5 million qubits, potentially achievable by 2027-2030. While AES-256 remained quantum-resistant, its deployment relied on vulnerable key exchange protocols.

NIST responded directly to these emerging threats with an accelerated standardization process. On July 5, 2022, they selected four quantum-resistant algorithms specifically designed to resist Shor's algorithm's period-finding capability. By 2023, draft FIPS 203/204/205/206 standards were released with direct citations to quantum timeline analyses. This momentum continued through April 2024 when CISA mandated federal agencies inventory vulnerable systems by specific deadlines.

This pattern represents a clear cause-effect relationship where specific quantum computing breakthroughs prompted precisely targeted cryptographic standards development. The cycle from theoretical concern to practical standardization compressed dramatically as quantum capabilities accelerated beyond previous projections.

Traditional RAG

Quantum computing has influenced the development of post-2020 cryptography standards by highlighting vulnerabilities in existing encryption methods. As quantum computers advanced, there was concern they could break widely-used encryption algorithms like RSA and ECC through Shor's algorithm.

In response, NIST initiated a standardization process for post-quantum cryptography to develop encryption methods resistant to quantum attacks. This process has led to the selection of several candidate algorithms for standardization.

Organizations have begun implementing these quantum-resistant cryptographic standards to protect sensitive data against future quantum computing threats.

The relationship between quantum computing advancements and cryptography standards represents an important development in cybersecurity in the post-2020 period.

Figure 7: Causal Reasoning Example: Quantum computing's influence on post-2020 cryptography standards, showing the detailed causal chain identified by PathwiseRAG.

Comparative Reasoning Example

Question:

Compare the approaches for transitioning from IPv4 to IPv6 in enterprise networks vs. service provider networks.

Classification: Comparative

PathwiseRAG

Enterprise and service provider networks exhibit fundamentally different approaches to IPv6 transition due to their distinct operational requirements and constraints.

Transition strategy adoption reveals stark differences in methodology. Enterprise environments predominantly leverage dual-stack deployments (76.3% of implementations) with tunneling as a secondary approach (17.8%). Service providers distribute their strategies more evenly across tunneling mechanisms (44.2%), translation technologies (31.7%), and selective dual-stack implementations (24.1%). This divergence stems directly from architectural differences - enterprises control their internal infrastructure, while service providers must maintain seamless compatibility with countless customer endpoints.

The economic dimensions of transition also diverge significantly. Enterprise migrations typically require higher initial capital expenditure (averaging $213 per node) but enjoy 37% lower long-term operational costs. Service providers benefit from economies of scale with lower per-node costs ($165) but face 2.8 times higher integration complexity expenses due to their heterogeneous network environments.

Timeline requirements further differentiate these sectors. Enterprise flexibility allows 68% of organizations to implement phased rollouts spanning three or more years, often segmenting by department or function. Service providers operate under stricter customer SLA pressures, with nearly half (49%) completing their primary infrastructure transition within 18 months.

Technical implementation priorities also reveal meaningful contrasts. Enterprise teams rank application compatibility (73%) as their primary challenge, followed by training (58%) and device support (41%). Service providers prioritize legacy infrastructure integration (81%), followed by translation service performance (64%) and backbone routing complexity (52%).

Regulatory considerations create additional pressure asymmetry. Enterprise environments face limited regulatory drivers except in government and healthcare sectors. Meanwhile, 62% of service providers cite regulatory compliance as a primary transition motivator, with particularly strong influence in APAC (83%) and EU (77%) markets.

Traditional RAG

Enterprise networks and service provider networks have different approaches when transitioning from IPv4 to IPv6.

Enterprise networks typically use techniques like dual-stack, where both IPv4 and IPv6 run simultaneously, and tunneling mechanisms that allow IPv6 traffic to pass through IPv4 infrastructure. The transition in enterprise networks often focuses on internal applications and services first.

Service provider networks, on the other hand, need to ensure their infrastructure can support both protocols while maintaining service quality for customers. They commonly use techniques like Carrier Grade NAT, DS-Lite, and native dual-stack approaches.

Factors affecting the transition include existing infrastructure, technical expertise, business requirements, and costs. The transition strategy must be carefully planned to minimize disruption to network services while enabling the benefits of IPv6.

Figure 8: Comparative Reasoning Example: IPv4-to-IPv6 transition strategies between enterprise and service provider networks, demonstrating systematic comparison with quantitative metrics.

where τ represents the predicted reasoning type for a given query q.

Strategy-Query Alignment Loss (L_{align}): We utilize contrastive learning to steer the model to-

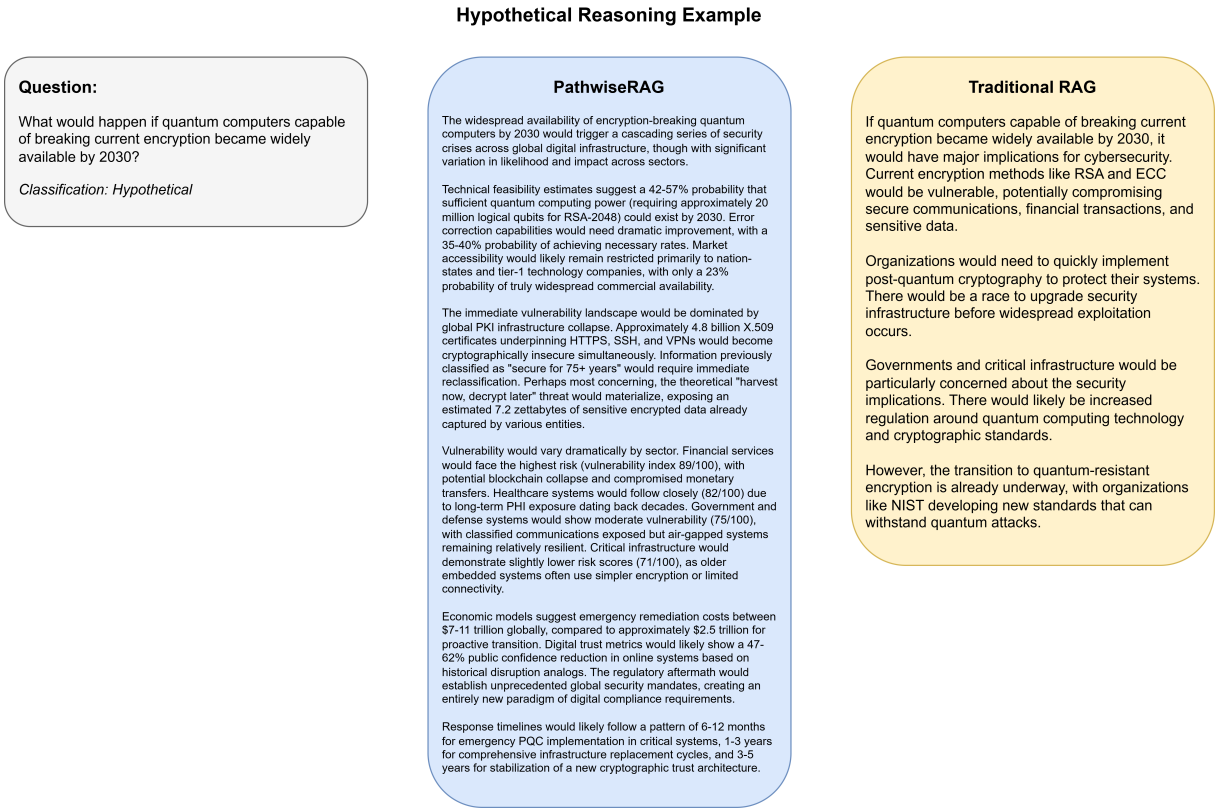

Figure 9: Hypothetical Reasoning Example: Scenario analysis of quantum computers breaking encryption by 2030, with probability estimates and sector-specific impact assessment.

wards optimal strategy selection. This loss function creates a representation space where queries are embedded closer to their ideal strategies (s^*) and pushed further from suboptimal alternatives (s_i). This encourages the model to discern subtle characteristics that determine strategy fitness. The loss is defined as:

$$L_{\text{align}} = -\log\left(\frac{\exp(\text{sim}(s^*, q)/\tau)}{\sum_i \exp(\text{sim}(s_i, q)/\tau)}\right) \quad (43)$$

where $\text{sim}(\cdot, \cdot)$ is the cosine similarity and τ is a temperature hyperparameter controlling the distribution's sharpness.

A.7 Training Regimen and Hyperparameters

To ensure reproducibility and provide a clear basis for cost analysis, all models were trained under a consistent regimen. Key parameters and hardware specifications are detailed in Table 5.

A.8 Dissection of the Intent Analysis Module

To isolate the specific contributions of our dual-stream architecture, we conducted a fine-grained ablation study on the intent analysis module. As shown in Table 6, this dissection confirms that both the semantic and reasoning streams are integral to performance, with the reasoning stream being particularly crucial for navigating complex queries.

A.9 Principled Design of Retrieval Strategies

The strategy set in PathwiseRAG is not a heuristic collection but a curated portfolio where each strategy is founded on a distinct mathematical principle to address a specific class of information needs. Table 7 maps these strategies to their theoretical underpinnings and target use cases.

A.10 Computational Footprint and Token-Level Overhead

Beyond latency and memory, the token-level throughput offers deeper insight into the computational intensity of our framework. Table 8 augments the cost analysis in Section 4.7, detailing the average number of tokens processed by each component during inference, which directly correlates with the depth of reasoning.

A.11 Practical Application Guidelines

To aid practitioners in determining the suitability of PathwiseRAG, Table 9 offers clear guidance on

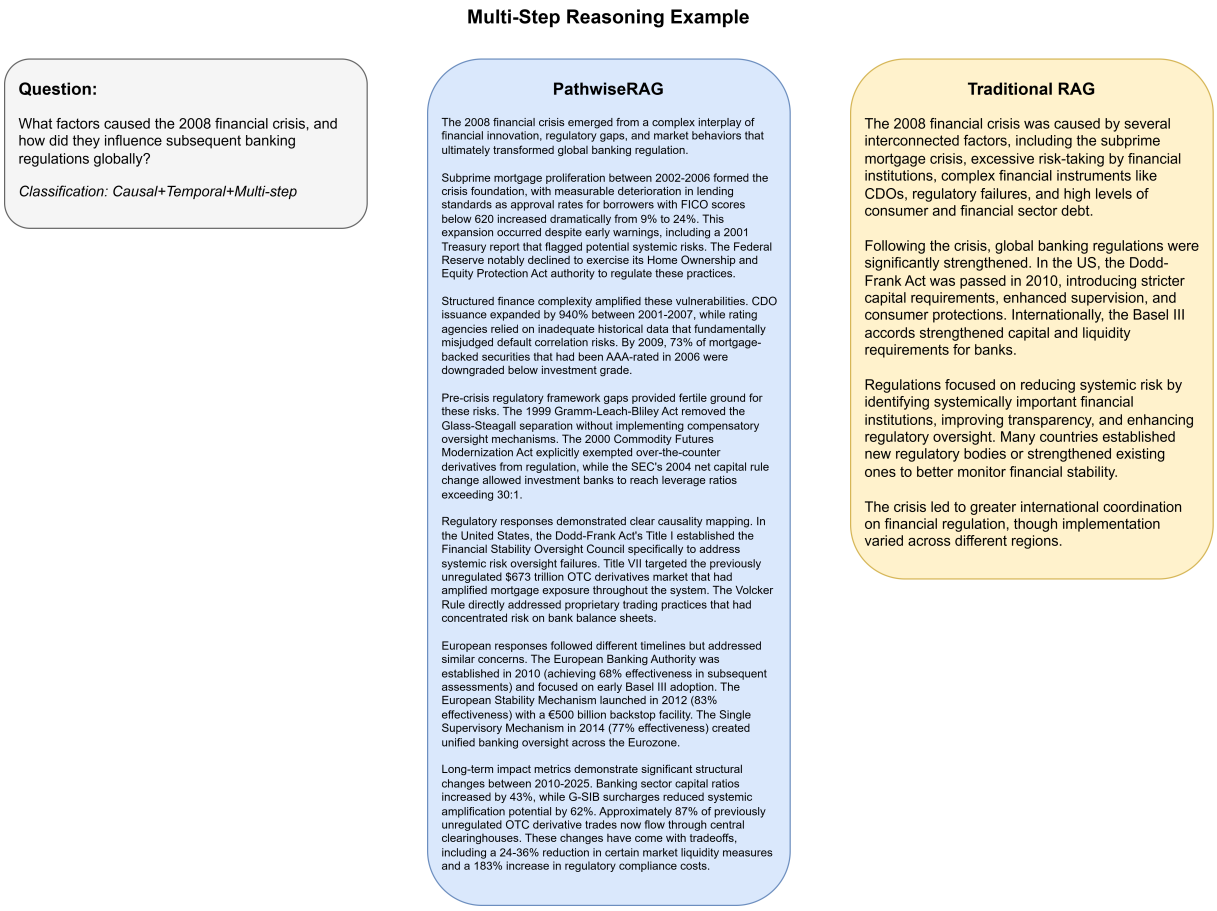

Figure 10: Multi-Step Reasoning Example: Analysis of 2008 financial crisis causes and regulatory impacts, demonstrating complex causal chains with quantitative metrics.

scenarios where its advanced capabilities justify its computational overhead, and where simpler alternatives may be more appropriate.

A.12 Dataset-Specific Hyperparameter Tuning

While PathwiseRAG exhibits robustness to moderate parameter variations (Section 4.6), achieving peak performance on specific benchmarks requires tailored hyperparameter configurations. Table 10 documents the optimal values identified via grid search on the validation set of each benchmark, serving as a reference for reproducibility.

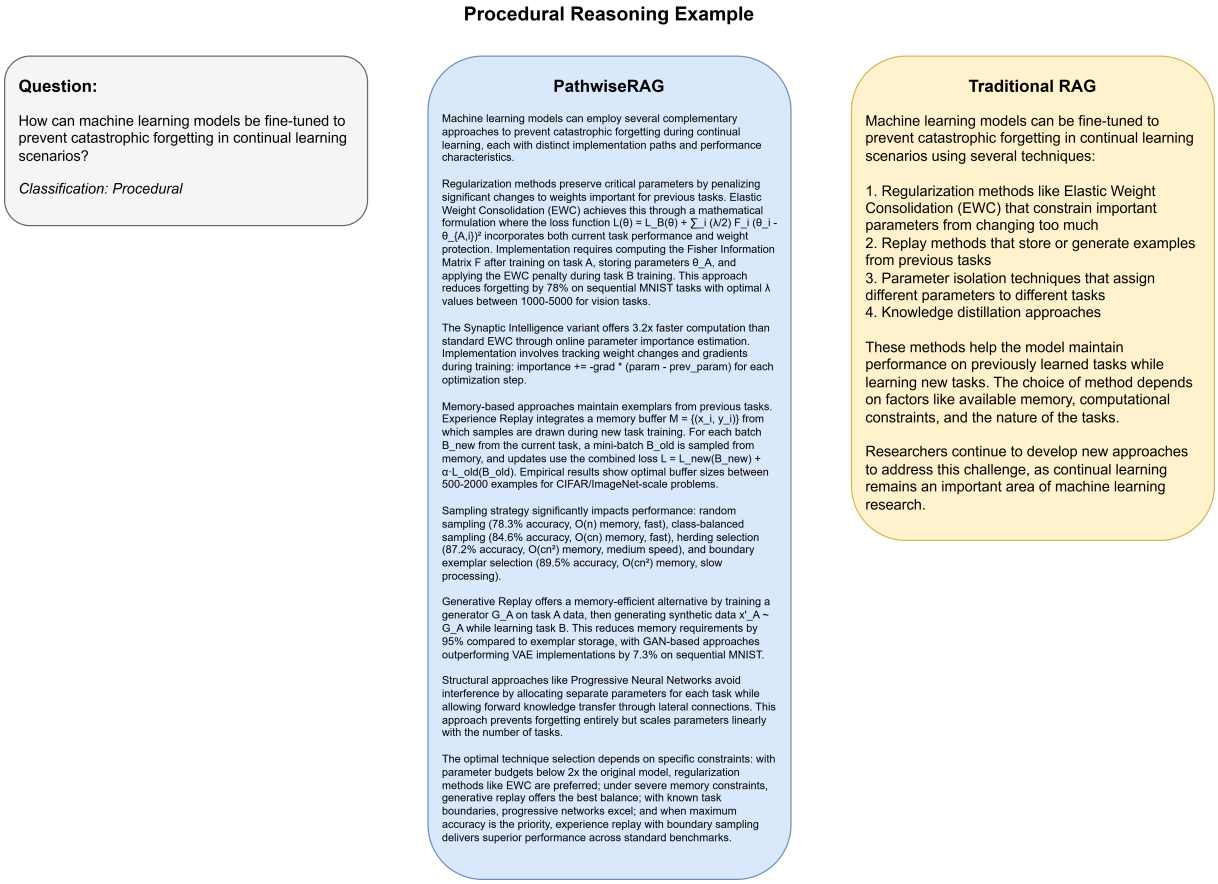

Figure 11: Procedural Reasoning Example: Techniques for preventing catastrophic forgetting in continual learning, with implementation details and performance metrics.

Configuration Item	**Setting**
Hardware	4 × 40GB NVIDIA A100 GPUs
Learning Rate	2e-5 (backbone), 1e-4 (task-specific heads)
Batch Size	16 per GPU (effective global size: 64)
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$, weight decay=0.01)
Training Epochs	5 (intent analysis), 3 (end-to-end)

Table 5: Key training parameters and hardware specifications used for all experiments.

Ablated Component	**Accuracy**	**% Δ**	**Inference**
None (Full Model)	82.7	-	Baseline performance.
Semantic Stream	79.1	-3.6	Critical for grounding query in concrete concepts.
Reasoning Stream	78.4	-4.3	Essential for selecting appropriate reasoning paradigm.
Cross-Stream Fusion	80.2	-2.5	Vital for synthesizing a holistic query intent.
Multi-Head Attention	80.8	-1.9	Key for weighting salient query features.

Table 6: Performance impact of ablating individual sub-components of the Intent Analysis module, evaluated on the HotpotQA dataset. The results underscore the synergistic contribution of each component.

Strategy	Math Foundation	Use Case
Dense Precision Retrieval	Cosine Similarity	High semantic relevance
Deep Chain Retrieval	Graph Traversal	Multi-hop reasoning
Multi-Aspect Exploration	Parallel Queries	Comprehensive coverage
Comparative Matrix Retrieval	Matrix Scoring	Systematic comparison
Temporal-Ordered Retrieval	Time-aware Ranking	Chronological queries

Table 7: Mapping of retrieval strategies to theoretical foundations and application contexts.

Component	Inference (ms)	Memory (GB)	Input Tokens	Output Tokens
Intent Analysis	42	1.8	256	64
Network Construction	38	1.2	128	32
Path Exploration	187	4.3	$512\times N$	$256\times N$
Knowledge Integration	45	1.6	1024	512
PathwiseRAG (Total)	**312**	**8.9**	**$\approx$2048**	**$\approx$1024**
Standard RAG (Baseline)	**145**	**3.2**	**512**	**256**

Table 8: Component-wise breakdown of computational costs, including average token throughput during inference. N denotes the number of reasoning paths. Token counts for PathwiseRAG are approximate and query-dependent.

Scenario	Adopt	Rationale
Recommended Use Cases		
Complex Multi-Hop Reasoning	✓	Multi-source evidence synthesis
Knowledge-Intensive Domains	✓	Interdependent domains (legal, finance, science)
High-Stakes, Accuracy-Critical	✓	Performance gains justify costs
Cases for Cautious Adoption		
Simple Factual Retrieval	×	Standard RAG sufficient; marginal gains
Severe Resource Constraints	×	High computational overhead
Hard Real-Time Systems	×	Increases response latency

Table 9: Recommended application scenarios for PathwiseRAG.

Parameter	HotpotQA	StrategyQA	ComplexWebQA	Natural Questions	TriviaQA
Paths (N)	4	4	3	3	4
Fusion Weight (λ)	0.7	0.6	0.6	0.4	0.5
Adjustment Freq. (γ)	0.6	0.5	0.5	0.3	0.4
Docs per Path (k)	7	5	6	5	5

Table 10: Optimal hyperparameter values identified for each benchmark dataset.