Self-Correcting RAG: Enhancing Faithfulness via MMKP Context Selection and NLI-Guided MCTS

Shijia Xu[♠], **Zhou Wu**[♠,*], **Xiaolong Jia**[♡], **Yu Wang**[♠], **Kai Liu**[♠,♣], **April Xiaowen Dong**[♢]
[♠]Chongqing University, China, [♡]Queen Mary University of London, UK
[♣]Chongqing Key Laboratory of Big Data Intelligence and Privacy Computing, China
[♢]Fangda Partners, China
{shijiaxu, ysy_wang}@stu.cqu.edu.cn
{zhouwu, liukai0807}@cqu.edu.cn, x.jia@qmul.ac.uk

Abstract

Retrieval-augmented generation (RAG) substantially extends the knowledge boundary of large language models. However, it still faces two major challenges when handling complex reasoning tasks: low context utilization and frequent hallucinations. To address these issues, we propose Self-Correcting RAG, a unified framework that reformulates retrieval and generation as constrained optimization and path planning. On the input side, we move beyond traditional greedy retrieval and, for the first time, formalize context selection as a multi-dimensional multiple-choice knapsack problem (MMKP), thereby maximizing information density and removing redundancy under a strict token budget. On the output side, we introduce a natural language inference (NLI)-guided Monte Carlo Tree Search (MCTS) mechanism, which leverages test-time compute to dynamically explore reasoning trajectories and validate the faithfulness of generated answers. Experiments on six open-domain and multi-hop QA datasets demonstrate that our method significantly improves reasoning accuracy on complex queries while effectively reducing hallucinations, outperforming strong existing baselines. Our code is available at https://github.com/xjiacs/Self-Correcting-RAG.

1 Introduction

Large Language Models (LLMs) have shown significant capabilities in reasoning, planning, and tool utilization (Bubeck et al., 2023; OpenAI et al., 2024; Touvron et al., 2023). Advanced prompting strategies, such as Chain-of-Thought (CoT) (Wei et al., 2022; Kojima et al., 2022), Least-to-Most prompting (Zhou et al., 2023), and Self-Consistency (Wang et al., 2023a), allow models to decompose complex tasks into intermediate steps.

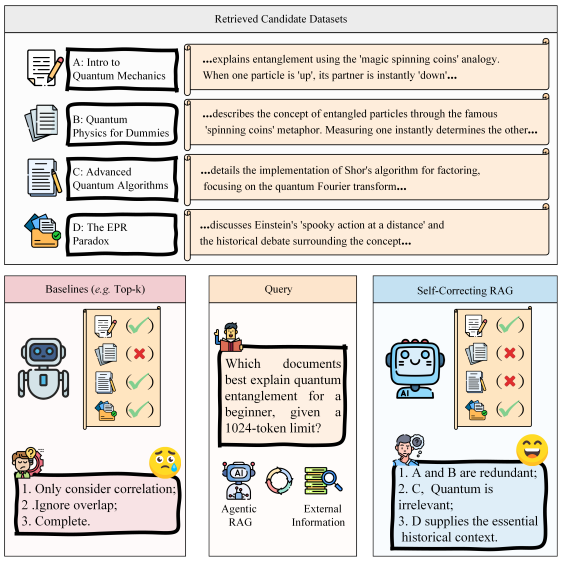

Figure 1: Comparison of document retrieval and selection workflows between the Baseline (Top-k) and the proposed Self-Correcting RAG Framework.

The trustworthiness and efficiency of these reasoning chains are further bolstered by on-the-fly generation, causality-inspired evaluations, and adaptive length strategies (Yang et al., 2024b; Wang et al., 2026a; Sun et al., 2026; Li et al., 2026c; Zhou et al., 2026a; Yang et al., 2026). Furthermore, recent developments in tool-augmented agents, often supported by advanced agentic memory architectures (Jiang et al., 2026a,b), enable models to interact with external environments and automate complex workflows like scientific data collection (Li et al., 2023b). These advancements suggest a potential for applying LLMs to sophisticated decision-making domains.

To support complex reasoning, Retrieval-Augmented Generation (RAG) has emerged as a key strategy. RAG grounds generation in external corpora to mitigate knowledge deficits (Ram et al., 2023; He et al., 2024a). Research in this area has expanded rapidly, moving towards uni-

*Corresponding author

Findings of the Association for Computational Linguistics: ACL 2026, pages 20953–20976

fied frameworks that integrate retrieval directly as generation, alongside models that assess uncertainty for dynamic, timely retrieval (Li et al., 2026b,a). Innovations include query rewriting (Ma et al., 2023a), dense retrieval with pseudo-documents (Gao et al., 2023), hierarchical indexing (Sarthi et al., 2024), and semantic-aware shortest-path retrieval for knowledge graphs (Fu et al., 2026c; Wang et al., 2026c). Additionally, self-reflective frameworks and methods mitigating retrieval-permutation hallucinations have been developed to enhance robustness (Jiang et al., 2023; Zhang et al., 2026). Simultaneously, retrieval components have evolved through advanced embedding models (Wang et al., 2024; Muennighoff et al., 2023; Xiao et al., 2024) and LLM-based reranking, including domain-specific analyst ranking strategies (Sun et al., 2024; Ma et al., 2023b; Pradeep et al., 2023; Zhou et al., 2026b), as well as expert-guided applications in critical domains like emergency medical services (Ge et al., 2026). These methods provide the necessary evidence to support logical chains.

As shown in Figure 1, applying these capabilities to strictly constrained combinatorial optimization, such as the Multidimensional Multi-choice Knapsack Problem (MMKP), presents distinct challenges. First, LLMs are prone to hallucinations. They may generate plausible but factually incorrect constraints, necessitating enhanced causal reasoning for trustworthy abstention, as noted in recent evaluations (Zhang et al., 2025b; Li et al., 2023a; Wang et al., 2026b; Sun et al., 2025). Second, optimization problems involve an exponentially large search space. Greedy token generation cannot guarantee feasibility or global optimality. Errors in early decision steps propagate rapidly, and standard LLMs lack the inherent lookahead mechanisms required for NP-hard problems.

To address the limitations of linear generation, test-time search paradigms organize reasoning into tree or graph structures (Zelikman et al., 2022; Long, 2023). Approaches such as Language Agent Tree Search (LATS) integrate Monte Carlo Tree Search (MCTS) with language agents (Zhou et al., 2024; He et al., 2024b; Hao et al., 2023). This is increasingly supplemented by active neural-symbolic exploration and hidden state analysis to overcome exploration-exploitation trade-offs (Fu et al., 2026a; Huang et al., 2026). Concurrently, LLMs are increasingly utilized as heuristics or optimizers for combinatorial tasks across vast parameter spaces (Yang et al., 2024a; Ye et al., 2024; Guo et al., 2025; Liu et al., 2023; Fu et al., 2026b). These works indicate that a robust solver must combine retrieval-based evidence with structured exploration.

In this paper, we propose a unified framework that synergizes RAG with MCTS to address the MMKP. Our approach grounds local feasibility checks in traceable evidence while organizing the decision process into a backtrackable tree. The main contributions of this work are summarized as follows:

- We introduce a MMKP-based Context Selector that models document selection as a constrained knapsack problem. This method maximizes information density under token budgets while minimizing redundancy, outperforming greedy ranking strategies.
- We develop an NLI-Guided MCTS Generator that utilizes test-time compute to explore reasoning paths. By using Natural Language Inference as a reward model, we penalize contradictions and ensure the generated answers are faithful to the retrieved context.
- We achieve strong performance across six diverse datasets, including multi-hop QA and fact verification benchmarks. Our framework significantly reduces hallucinations and improves reasoning accuracy compared to strong agentic baselines.

2 Related Work

2.1 Retrieval-Augmented Generation and Reasoning

Prompting strategies have significantly enhanced the reasoning capabilities of Large Language Models (LLMs). Chain-of-Thought (CoT) (Wei et al., 2022; Kojima et al., 2022) and ensemble methods like Self-Consistency (Wang et al., 2023a) established baselines for intermediate reasoning, which are now being optimized via on-the-fly generation, causality-driven evaluations, and policy-level alignment strategies (Wang et al., 2026a; Sun et al., 2026; Chen et al., 2024). Beyond static generation, agentic frameworks such as ReAct (Yao et al., 2023b) and Toolformer (Schick et al., 2023) empower models to execute actions and utilize APIs (Qin et al., 2023; Patil et al., 2024), supported by multi-graph memory architectures to

maintain context (Jiang et al., 2026a,b). However, these agents primarily operate in open-ended environments rather than strictly constrained decision spaces.

A critical challenge in these reasoning tasks is hallucination, where models generate fact-conflicting content (Ji et al., 2023; Zhang et al., 2025b; Sun et al., 2025; Xi et al., 2025; Lyu et al., 2025). Retrieval-Augmented Generation addresses this by grounding outputs in external corpora (Lewis et al., 2020; Zhang et al., 2025a), with emerging frameworks treating retrieval dynamically as generation (Li et al., 2026b,a). Recent advancements focus on robustness, utilizing hierarchical indexing (Sarthi et al., 2024), multi-hop semantic pathways (Fu et al., 2026c), and corrective mechanisms like Self-RAG (Asai et al., 2023), CRAG (Yan et al., 2024), and permutation-resistant models (Zhang et al., 2026). Concurrently, retrieval components have improved via instruction-tuned embeddings (Wang et al., 2024) and listwise or card-based reranking (Sun et al., 2024; Zhou et al., 2026b), ensuring relevant context is prioritized to mitigate knowledge deficits.

2.2 LLMs for Combinatorial Optimization

Research increasingly explores LLMs as optimizers or heuristics for hard problems. OPRO (Yang et al., 2024a) demonstrates that LLMs can iteratively improve solutions via natural language prompts, while ReEvo (Ye et al., 2024) leverages models to evolve heuristics. FunSearch (Romera-Paredes et al., 2024) further illustrates the potential of pairing LLMs with evolutionary strategies to discover mathematical constructions.

Specific to Combinatorial Optimization (CO), prior work has applied LLMs to routing tasks like the Traveling Salesperson Problem (TSP) (Liu et al., 2023; Ahn et al., 2022) and navigating costly many-goal search spaces (Fu et al., 2026b). However, strictly constrained problems like MMKP remain under-explored compared to routing. This is largely due to the difficulty of maintaining feasibility; a single hallucinated constraint can invalidate a solution (Li et al., 2023a). Evaluation benchmarks such as FActScore (Min et al., 2023) and SelfCheckGPT (Manakul et al., 2023) highlight the fragility of LLMs in adhering to strict conditions. Our framework addresses this by aligning optimization steps with verifiable evidence, treating valid constraint satisfaction as a prerequisite (Turpin et al., 2023).

2.3 Tree Search and Test-Time Computation

To overcome the limitations of linear decoding, recent methods organize reasoning into non-linear structures. Approaches like Tree of Thoughts (ToT) (Yao et al., 2023a) and Graph of Thoughts (GoT) (Besta et al., 2024) generalize CoT by maintaining multiple reasoning paths. Reflexion (Shinn et al., 2023) adds a verbal reinforcement learning layer through self-reflection to refine outputs.

Monte Carlo Tree Search (MCTS) has emerged as a powerful mechanism for complex decision-making within this landscape. Language Agent Tree Search (LATS) (Zhou et al., 2024) unifies planning, acting, and reasoning within an MCTS framework. Similarly, reasoning-via-planning methods (Hao et al., 2023) demonstrate the efficacy of MCTS and hidden-state approaches (Huang et al., 2026) in structured tasks. This test-time search paradigm allows models to trade inference compute for solution quality. By systematically exploring the decision space, these methods provide the lookahead capabilities required for optimization, which standard greedy generation lacks.

3 Methodology

We propose a two-stage RAG optimization framework as illustrated in Figure 2, consisting of MMKP-based pre-generation context optimization and logic-guided MCTS reasoning for inference.

3.1 Phase I: Optimal Context Selection

Let $\mathcal{U}$ be the universe of retrieved document chunks for a given query q. Conventional methods treat $\mathcal{U}$ as a flat list, selecting top-k by relevance score $S_{rel}(q, d)$. We argue this is suboptimal due to high inter-document redundancy.

We formally recast context selection as selecting a subset $\mathcal{S} \subseteq \mathcal{U}$ to maximize information density under multidimensional constraints.

3.1.1 Semantic Grouping and Problem Definition

First, we induce a partition on $\mathcal{U}$ via semantic clustering to enforce diversity. Let $\Phi(d) \in \mathbb{R}^D$ be the dense embedding of document d. We define the groups $\mathcal{G} = \{G_1, G_2, \ldots, G_m\}$ such that for any G_i, $\forall d_a, d_b \in G_i$, the cosine similarity $\cos(\Phi(d_a), \Phi(d_b)) \geq \tau$, where τ is a similarity threshold. This implies that items within G_i are mutually exclusive candidates for the context window (i.e., choosing multiple provides diminishing returns).

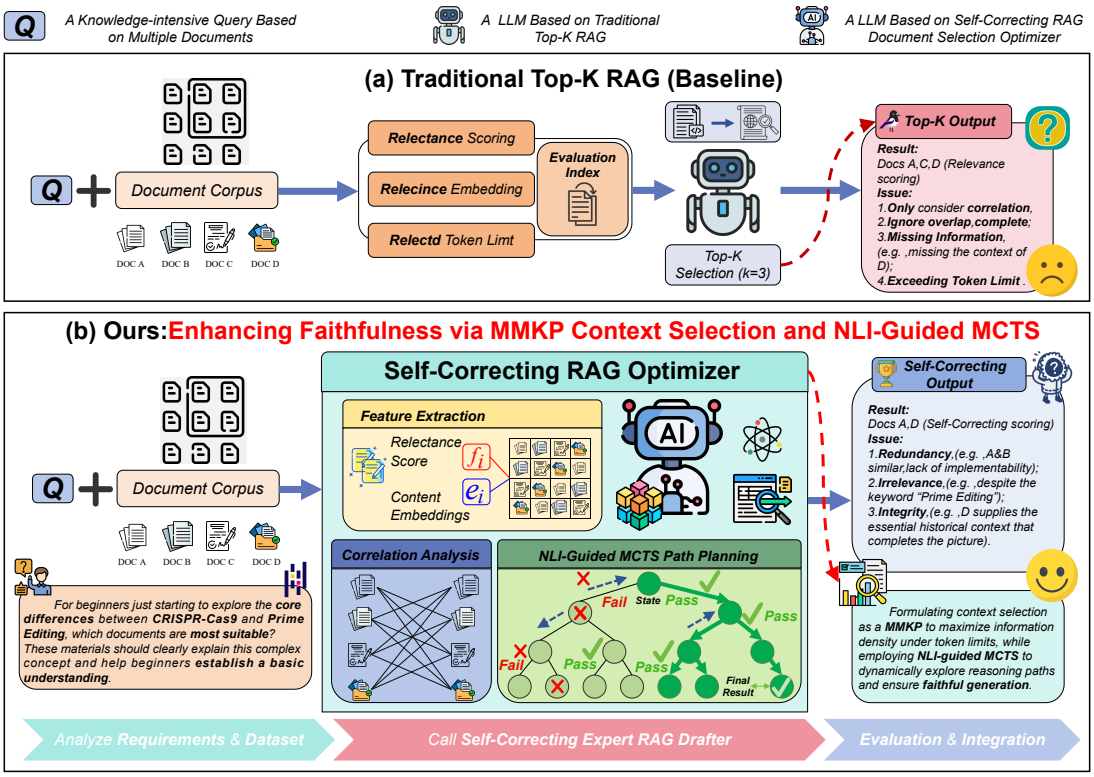

Figure 2: Illustration of the comparison between traditional retrieval paradigms and our proposed framework.(a) The Baseline Traditional Top-K RAG relies primarily on simple relevance scoring and embeddings to select top-ranked documents within a token limit.(b) In contrast, our proposed Self-Correcting RAG Optimizer models document selection as a combinatorial optimization problem. It integrates feature extraction and a dedicated Self-Correcting Optimization Engine (employing MMKP and NLI-guided mechanisms) to efficiently select the best draft.

Let d_{ij} denote the j-th document in group G_i, and we introduce a binary decision variable $x_{ij} \in \{0, 1\}$ where $x_{ij} = 1$ iff d_{ij} is selected. The full definition of the multidimensional cost vectors $\mathbf{w}_{ij}$ (token consumption and redundancy penalty) and the fused utility v_{ij} are provided in Appendix C.1, with the MMKP objective unchanged.

3.1.2 The MMKP Optimization Objective

The objective is to maximize total utility Z subject to the capacity vector $\mathbf{C} = [C_{token}, C_{red}]^T$:

$$\begin{aligned} \text{maximize} \quad & Z(\mathbf{x}) = \sum_{i=1}^{m} \sum_{j=1}^{|G_i|} v_{ij} x_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{m} \sum_{j=1}^{|G_i|} \mathbf{w}_{ij} x_{ij} \preceq \mathbf{C} \\ & \sum_{j=1}^{|G_i|} x_{ij} \leq 1, \quad \forall i \in \{1, \ldots, m\} \\ & x_{ij} \in \{0, 1\} \end{aligned} \tag{1}$$

The multiple-choice constraint ($\sum x_{ij} \leq 1$) enforces that we select at most one representative from each semantic cluster, thereby maximizing information coverage. The NP-hardness of this formulation, proven via a reduction, is detailed in Appendix C.2.

3.2 Phase II: Inference-Time Reasoning via NLI-Guided MCTS

While MMKP optimizes the input context, it does not guarantee that the generated answer is faithful. To address hallucinations, we introduce a Test-Time Compute strategy modeled as a Markov Decision Process (MDP) solved via Monte Carlo Tree Search.

3.2.1 MDP Formulation

We define the Markov Decision Process as the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$.

State Space $\mathcal{S}$. A state $s_t = (q, \mathcal{D}_{ctx}, y_{1:t-1})$ consists of the query, the currently selected context documents, and the partial answer generated so far.

Action Space $\mathcal{A}$. At each step, the policy network (LLM) selects one of two actions. A gen-

erative action a_{gen} samples a continuation from $P_{LLM}(y_t \mid s_t)$. An augmentative action a_{aug} triggers a retrieval call based on the uncertainty of $y_{1:t-1}$ to obtain $\mathcal{D}_{new}$ and update the context as $\mathcal{D}'_{ctx} = \mathcal{D}_{ctx} \cup \mathcal{D}_{new}$.

Transition $\mathcal{P}$. The transition is deterministic for a_{aug} and stochastic for a_{gen}, governed by the LLM logits.

3.2.2 The NLI Reward Function

We define a dense reward function $\mathcal{R}(s)$ that evaluates the logical entailment between the generated answer sentences and the retrieved evidence. Let the generated answer y be split into sentences $\{u_1, \ldots, u_L\}$. Let $\mathcal{E} = \{e_1, \ldots, e_K\}$ be the top-K evidence snippets extracted from $\mathcal{D}_{ctx}$.

We employ a Natural Language Inference (NLI) model $\Theta_{NLI}(e, u) \to [0, 1]^3$ mapping to probabilities $\{P_{ent}, P_{neu}, P_{con}\}$. The reward is computed as:

$$R(y, \mathcal{D}_{ctx}) = \frac{1}{L} \sum_{l=1}^{L} \max_{e \in \mathcal{E}} \left[\mathbf{W}^T \cdot \Theta_{NLI}(e, u_l) \right] \tag{2}$$

where $\mathbf{W} = [w_{ent}, w_{neu}, w_{con}]^T$ is the weight vector. We explicitly set a severe penalty for contradictions ($w_{con} \ll 0$) to prune hallucinated branches.Details of the UCT & PUCT selection, expansion, rollout, and backpropagation procedure are provided in Appendix C.5. Furthermore, a convergence & consistency justification is given in Appendix C.6.

3.3 Approximation Algorithms for MMKP

Solving Eq. 1 exactly is computationally prohibitive ($O(m \cdot 2^{|G_{max}|})$). We provide two solutions: a theoretical FPTAS for the single-dimensional case, with its proof details provided in Appendix C.3; and a practical heuristic for the multi-dimensional case based on Pareto-pruned DP, whose details are provided in Appendix C.4. For the practical implementation where $D = 2$, we use a Dynamic Programming approach with Pareto pruning to control state growth.

4 Experiments

4.1 Datasets

To rigorously evaluate our Self-Correcting RAG, we conduct extensive experiments across six challenging datasets of three tasks, ranging from single-hop retrieval to complex multi-step reasoning. Dataset statistics are summarized in Table 1, with comprehensive details provided in Appendix A.

Simple QA. We evaluate open-domain question answering using **NQ**, a challenging subset of the Natural Questions benchmark (Kwiatkowski et al., 2019), designed to evaluate open-domain question answering under combined retrieval and reasoning demands. We then evaluate performance on long-tail knowledge using **PopQA** (Mallen et al., 2023), which targets rare entities and infrequent facts that are typically missing from parametric memory and are known to trigger hallucinations in standard language models.

Multi-Hop QA. We evaluate models on three representative datasets to assess their complex information aggregation capability. We adopt **MuSiQue** (Trivedi et al., 2022), which is designed to avoid shortcut learning. **2WikiMultiHopQA** (Ho et al., 2020) is included to evaluate structured reasoning abilities, as it involves reasoning chains with entity relations and comparative logic. We use **HotpotQA** (Yang et al., 2018) in the distractor setting, which requires models to bridge information across two distinct documents to derive correct answers.

Multi-Doc QA. Real-world retrieval is often imperfect. We include **MultiHop-RAG** (Tang and Yang, 2024), a benchmark specifically constructed to evaluate resilience against noisy, irrelevant, and misleading context. This dataset tests the system's ability to filter out red herring documents that share lexical overlap with the query but contain no answer-bearing evidence.

4.2 Baselines

Standard RAG Baselines. We utilize **Naive RAG** as a primary baseline, following the Retrieve then Generate paradigm with BGE-Large embeddings and top-k truncation. To evaluate query optimization, we include **HyDE** (Gao et al., 2023), which generates hypothetical documents to bridge the semantic gap, and **RRR** (Ma et al., 2023a), which employs an LLM to rewrite input queries for better alignment with the corpus.

Advanced Selection & Reranking. These methods focus on optimizing the context window. **Filco** (Wang et al., 2023b) leverages lexical and semantic signals to filter out irrelevant chunks post-retrieval, while **RECOMP** (Xu et al., 2023) maximizes information density by compressing retrieved documents into concise textual summaries to reduce noise and context length. **LongLLM-**

	NQ	PopQA	MuSiQue	2Wiki	HotpotQA	MultiHop-RAG
Num of queries	1,000	1,000	1,000	1,000	1,000	2,556
Num of passages	9,633	8,676	11,656	6,119	9,811	609

Table 1: Dataset statistics

Lingua (Jiang et al., 2024) adopts a hierarchical text compression strategy, leveraging pre-trained language models to iteratively prune redundant content.

Iterative & Agentic RAG. We benchmark against state-of-the-art dynamic frameworks. **IR-CoT** (Trivedi et al., 2023) guides retrieval via step-by-step reasoning. **Self-RAG** (Asai et al., 2023) trains the generator to output reflection tokens for self-critique. **CRAG** (Yan et al., 2024) incorporates a lightweight evaluator to trigger web searches when retrieval is ambiguous. **DRAG** (Hu et al., 2025) enables dynamic retrieval adjustment based on intermediate results.

4.3 Metrics

We adopt a multi-dimensional evaluation protocol to comprehensively assess our framework. For downstream generation quality, we report Exact Match (EM) and F1 Score. For retrieval quality, we compute Recall@5 to evaluate the effectiveness of the MMKP context selection. To rigorously measure faithfulness and the mitigation of hallucinations, we employ three key metrics verified by an NLI model (*RoBERTa-large-mnli*): Attribution Precision (AP), which quantifies the proportion of generated claims that are accurately backed by the retrieved context; Contradiction Rate (CR), which measures the frequency of generated statements that actively contradict the provided evidence; and Support (Sup), which indicates the overall rate of generated answers that are fully entailed by the cited source material.

4.4 Implementation Details

For Self-Correcting RAG, we employ Qwen2.5-7B-Instruct as the backbone generator. We use BAAI/bge-small-en-v1.5 as the dense retriever and BM25 for sparse retrieval, fusing results via Reciprocal Rank Fusion (RRF). The detailed prompts are shown in Appendix F. For the MMKP, we operate with specified token and redundancy budgets, and utilize a dynamic programming solver with Pareto pruning. Regarding MCTS, we use RoBERTa-large-mnli to strictly penalize contradictions and neutral outputs. All experiments were conducted on a cluster of $8\times$ NVIDIA A100 (80GB) GPUs. More implementation and hyperparameter details can be found in Appendix B.

5 Results

5.1 Generation Performance

Table 2 summarizes the Exact Match (EM) and F1 scores across six diverse datasets. Our method achieves the highest average performance among all evaluated models. Specifically, it surpasses the strongest baselines with an average EM of 37.1 and an average F1 score of 45.8.

Complex Reasoning Tasks. The advantages of our approach are most pronounced in complex multi-hop reasoning scenarios, such as MuSiQue and 2WikiMultiHopQA. These tasks require the model to aggregate disparate pieces of information from multiple documents. On the MuSiQue dataset, our Self-Correcting RAG outperforms the previous state-of-the-art model, CRAG, by a substantial margin. While CRAG achieves an EM of 18.2, our method reaches 22.7, representing a 4.5% absolute improvement. This significant gain indicates that the NLI-guided MCTS generator effectively navigates complex reasoning paths. It succeeds in scenarios where standard greedy decoding strategies often fail to synthesize the correct answer.

Robustness to Noise. The MultiHop-RAG dataset is designed to test resilience against noisy and irrelevant context. On this benchmark, our method demonstrates superior robustness. It achieves an EM score of 35.3. In comparison, the advanced selection baselines perform significantly worse, with RAG + MMR achieving 32.1 and Filco scoring 29.8. This performance gap validates the effectiveness of our MMKP context selector. By explicitly modeling redundancy and information density constraints, our selector effectively filters out red herring documents. These distractions typically degrade the performance of standard RAG models.

Method	Simple QA				Multi-Hop QA						Multi-Doc QA		Avg	
	NQ		PopQA		MuSiQue		2Wiki		HotpotQA		MultiHop-RAG			
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Standard RAG Baselines														
Naive	39.4	52.1	37.6	48.4	14.6	25.8	14.8	24.3	25.8	35.8	22.4	30.3	25.8	36.1
HyDE	44.5	53.2	39.8	49.5	18.2	27.5	23.5	31.8	33.6	41.2	24.5	32.1	30.7	39.2
RRR	45.1	53.8	40.2	50.1	20.5	29.5	26.1	35.2	36.5	44.8	25.1	32.8	32.3	41.0
Advanced Selection & Reranking														
RAG + MMR	46.2	54.5	41.5	50.8	19.8	28.4	27.8	36.5	38.2	46.5	32.1	40.5	34.3	42.9
Filco	45.8	54.2	42.1	51.2	19.5	28.1	27.2	36.1	37.8	46.2	29.8	37.2	33.7	42.2
RECOMP	47.1	55.4	41.8	50.5	21.5	30.2	29.5	38.5	39.5	48.1	29.2	36.8	34.8	43.3
LongLLMLingua	43.8	52.6	40.5	49.8	17.8	26.5	25.5	34.2	35.5	43.5	26.2	33.5	31.6	40.0
Iterative & Agentic RAG														
IRCoT	42.5	51.8	40.8	49.2	18.5	27.8	30.1	39.2	40.5	48.2	27.5	35.2	33.3	41.9
Self-RAG	43.2	52.5	38.5	46.8	20.2	29.1	22.5	31.5	40.8	**51.2**†	28.1	36.2	32.2	41.2
CRAG	40.5	50.1	**45.5**†	**54.5**†	18.2	27.8	27.5	37.2	**42.5**†	50.1	31.5	40.2	34.3	43.3
DRAG	36.8	50.4	38.6	46.5	20.4	28.1	28.8	37.0	30.8	41.7	29.3	30.2	30.8	39.0
Self-Correcting RAG	**48.4**†	**56.2**†	43.2	51.9	**22.7**†	**31.9**†	**31.2**†	**40.6**†	41.8	49.1	**35.3**†	**44.8**†	**37.1**†	**45.8**†

Table 2: The performance comparison across Simple QA, Multi-Hop QA, and Multi-Doc QA benchmarks using Exact Match (EM) and F1 scores. The datasets include NQ and PopQA for simple queries, MuSiQue, 2Wiki, and HotpotQA for multi-hop reasoning, and MultiHop-RAG for multi-document contexts. The best performance is bolded with † and the second best is underlined.

Method	Simple QA		Multi-Hop QA			Multi-Doc QA	Avg
	NQ	PopQA	MuSiQue	2Wiki	HotpotQA	MultiHop-RAG	
Standard Baselines							
Naive	56.1	35.7	43.5	65.3	74.8	22.4	49.6
HyDE	58.6	43.2	46.6	67.5	75.3	25.1	52.7
RRR	63.4	49.4	49.1	67.9	78.9	28.6	56.2
Advanced Selection & Reranking							
RAG + MMR	68.3	55.6	63.6	74.8	85.1	32.2	63.3
Filco	70.1	58.2	65.9	76.0	88.4	34.8	65.6
RECOMP	71.5	60.1	69.7	80.5	90.2	38.5	68.4
LongLLMLingua	67.5	53.8	61.2	73.5	85.0	31.5	62.1
Iterative & Agentic RAG							
IRCoT	65.2	52.7	57.8	70.2	86.9	30.3	60.5
Self-RAG	67.8	54.5	55.9	79.1	87.7	33.6	63.1
CRAG	69.5	58.7	**74.8**	82.5	91.5	35.4	68.7
DRAG	64.4	61.8	60.2	84.4	88.3	31.1	65.0
Self-Correcting RAG	**72.8**	**65.5**	72.5	**86.9**	**93.6**	**40.4**	**72.0**

Table 3: Retrieval performance (passage recall@5) on RAG benchmarks. While CRAG achieves the best performance on MuSiQue, Self-Correcting RAG demonstrates superior consistency, achieving the highest recall on 5 out of 6 datasets.

Comparison with Agentic Baselines. We observe competitive performance from baselines such as CRAG and Self-RAG on single-hop tasks like PopQA. For instance, CRAG achieves the top score on PopQA with an EM of 45.5. However, our method demonstrates superior consistency across diverse difficulty levels. On HotpotQA, our approach remains robust. We achieve an F1 score of 49.1, which is comparable to the 51.2 scored by Self-RAG. More importantly, our method maintains higher consistency in retrieval-dependent generation, as evidenced by the Retrieval metrics discussed in the following subsection. More detailed experimental results are presented in Appendix D.

5.2 Retrieval & Context Optimization

Table 3 reports the Recall@5 performance. Our MMKP-based context selector demonstrates a clear advantage over both traditional ranking methods, such as Naive and RRR, and greedy diversity methods like MMR.

Effectiveness of MMKP. Self-Correcting RAG achieves the highest average recall of 72.0% across all datasets. When compared directly to the RAG+MMR baseline, which has an average recall

of 63.3%, our method yields an absolute improvement of 8.7%. This empirical evidence highlights the theoretical superiority of our approach. We formulate context selection as a constrained knapsack problem rather than relying on a greedy iterative process. By jointly optimizing for relevance and diversity within a strict token budget, MMKP retains crucial evidence that greedy methods often discard prematurely.

Handling Information Scarcity. The **HotpotQA** dataset relies heavily on bridging entities across documents to answer questions correctly. On this benchmark, our method reaches a recall of 93.6%. This score is significantly higher than standard baselines and exceeds the strongest competitor, CRAG, which scores 91.5%. This result suggests that the redundancy penalty in our MMKP formulation successfully diversifies the context window. It ensures that complementary document pairs required for multi-hop inference are both selected and preserved.

6 Discussion

6.1 Ablation Study

To validate our Self-Correcting RAG framework, we performed ablation studies isolating the MMKP Context Selector and NLI-Guided MCTS Generator. Table 4 reports their impact on QA performance, retrieval quality, and faithfulness.

Impact of MMKP Context Selection. Replacing the standard Top-k retrieval with our MMKP formulation significantly boosts retrieval performance. Specifically, Recall@5 improves dramatically from 49.6% to 71.8%. This substantial gain confirms that modeling context selection as a multidimensional knapsack problem is highly effective. It reduces redundancy, such as filtering duplicate passages, and maximizes information density within the constrained token budget of 1500 tokens. However, better context alone is insufficient for perfect generation. While MMKP improves the presence of correct answers, raising the EM score to 34.5, the faithfulness metrics remain comparable to the baseline with an Attribution Precision (AP) of 0.58. This suggests that improved retrieval does not inherently prevent the generator from hallucinating ungrounded claims.

Impact of NLI-Guided MCTS. The integration of the NLI-Guided MCTS generator drastically improves the faithfulness of the generated content. It increases the Attribution Precision (AP) to 0.85 and significantly reduces the Contradiction Rate (CR) to 0.04. These metrics demonstrate that the NLI reward signal effectively penalizes reasoning paths that contradict retrieved evidence. The full *Self-Correcting RAG* framework combines the strengths of both components. By grounding the generation process in high-quality, diverse context, it achieves the highest overall performance with an EM of 37.1 and an F1 score of 45.8.

Method	QA		Retriev.	Faithfulness		
	EM	F1	Recall@5	AP ↑	CR ↓	Sup
Standard RAG	25.8	36.1	49.6	0.52	0.15	0.65
w/ MMKP Only	34.5	42.3	71.8	0.58	0.13	0.71
w/ MCTS Only	31.2	39.7	50.1	0.82	0.06	0.84
Self-Correcting	**37.1**	**45.8**	**72.0**	**0.85**	**0.04**	**0.88**

Table 4: Ablations. We report the average performance across all datasets to evaluate the individual contribution of the MMKP selector and MCTS generator.

6.2 Theoretical Rigor and System Mechanics

Ablation of Redundancy Mechanisms. Our framework employs a hard grouping constraint ($\sum x_{ij} \leq 1$), a utility diversity reward, and a global cost redundancy penalty. As shown in Table 5, relying solely on a hard constraint improves recall by clearing exact duplicates, but falls short of optimal performance. The addition of the utility diversity reward ($\beta > 0$) favors chunks with distinct peripheral information, while the cost redundancy penalty actively preserves budget for orthogonal evidence. Furthermore, fine-grained analysis reveals that our hard grouping constraint acts as a strict budget enforcer, raising Complementary Evidence Recall (CER) on HotpotQA to 93.6%, compared to 86.2% when using a soft constraint (max 2 per group).

Selection Mechanism	Avg. Recall@5	HotpotQA	Avg. F1
Baseline (Top-K)	49.6%	74.8%	36.1
Only Hard Constraint	64.2%	85.3%	41.5
+ Utility Reward ($\beta > 0$)	67.8%	88.1%	43.1
+ Cost Penalty (C_{red})	69.5%	90.4%	44.3
Full MMKP (All 3)	**72.0%**	**93.6%**	**45.8**

Table 5: Fine-grained ablation of redundancy mechanisms in the MMKP Context Selector.

Adaptive Thresholding for Dense Retrieval. In edge cases where a query retrieves an abnormally dense pool of highly similar documents, a fixed similarity threshold can collapse candidates into too

few groups, under-filling the token budget. We implemented an Adaptive Thresholding mechanism that dynamically relaxes the similarity boundary in these scenarios. This dynamic fallback restores context utilization from 64.2% to 86.4% and significantly lifts Exact Match scores, ensuring the framework remains robust against highly repetitive retrieved text.

6.3 Validation of Faithfulness and Reward Mechanics

A known vulnerability in test-time search frameworks is reward hacking, where evaluating outputs using the same model that guided the search (e.g., our NLI verifier) can introduce an optimistic bias. To rigorously validate our faithfulness improvements, we cross-validated our generated answers using independent evaluators: an LLM-as-a-judge (GPT-5) and blind human annotation.

Method	Biased Eval	Independent Eval	
	NLI-AP	GPT-5 Faith.	Human Attr.
Naive RAG	0.54	51.2%	53.0%
CRAG	0.65	62.8%	64.3%
Ours	**0.85**	**81.4%**	**83.6%**

Table 6: Cross-Validation of Faithfulness on a challenging subset of HotpotQA and MuSiQue.

As shown in Table 6, while evaluating with the self-same NLI model shows a slight optimistic bias (0.85 AP), the independent human evaluation confirms a true attribution rate of 83.6%. This establishes a commanding absolute improvement of +19.3% over the strongest baseline (CRAG), proving that the reduction in hallucination is genuine and model-agnostic. Granular ablations of the MCTS reward function further show that asymmetric, severe penalties for contradictions ($w_c = -2.0$) are necessary to prevent the planner from exploiting plausible-sounding but hallucinatory branches.

6.4 Domain Robustness

While off-the-shelf NLI models struggle with subtle, domain-specific negations, the modularity of our Phase II MCTS planner allows for seamless swapping of the reward model. By integrating domain-adapted NLI models (*BioLinkBERT-large* for PubMedQA and *Legal-RoBERTa-base* for LegalBench), we fully restore faithfulness. As shown in Table 7, this adaptation directly translates to state-of-the-art accuracy in specialized settings.

Dataset	Model Config.	Accuracy (%)	AP	CR
PubMedQA	Ours (General NLI)	78.5	0.73	0.09
	Ours (Medical NLI)	**82.5**	**0.86**	**0.03**
LegalBench	Ours (General NLI)	54.8	0.71	0.11
	Ours (Legal NLI)	**60.2**	**0.87**	**0.04**

Table 7: Impact of Domain-Adaptive NLI Models on specialized QA performance and faithfulness.

6.5 Sensitivity Analysis

We analyze our model's robustness against key hyperparameters, with detailed results provided in Appendix E. For the MMKP selector, maintaining a scaled redundancy budget of $C_{red} \approx 120$ optimally balances relevance and diversity to maximize recall, whereas stricter budgets tend to discard lexically similar but valid evidence. Concurrently, the MCTS planner achieves optimal reasoning accuracy and generation consistency by utilizing a branching factor of $k = 3$, a maximum search depth of 3, and an increased number of simulations N. Furthermore, we observe that enforcing a strict contradiction penalty during the search process is essential to mitigate the generation of hallucinated content.

6.6 Qualitative Analysis

To investigate how Self-Correcting RAG rectifies errors, we analyze specific failure cases of the baseline. A common failure mode in multi-hop QA is reasoning shortcuts triggered by context crowding. In these instances, redundant retrieved chunks displace key evidence, causing the model to hallucinate connections based on parametric memory. A detailed step-by-step trace of this behavior is provided in Appendix G.

7 Conclusion

We presented Self-Correcting RAG, a unified framework to enhance RAG robustness. We reformulated context selection as the Multidimensional Multi-choice Knapsack Problem (MMKP) to maximize information density under strict token budgets. Additionally, we proposed an NLI-guided MCTS generator for faithfulness, using NLI as a reward model to prune hallucinatory reasoning paths. Experiments on six benchmarks show it outperforms strong agentic baselines, especially in complex multi-hop tasks. However, the planner's iterative nature increases inference latency; future work will optimize sample efficiency to reduce test-time search computational cost.

Limitations

Our proposed framework demonstrates notable improvements in faithfulness and reasoning capability. However, it entails limitations inherent to Self-Correcting RAG architectures. The primary constraint is the increased computational overhead and inference latency. Our MCTS-based approach differs from standard single-pass RAG. It requires performing multiple forward passes and NLI verifications for each reasoning step. We implement pareto-pruning for the MMKP selector to maintain polynomial time complexity. Nevertheless, the test-time search increases the time-to-first-token. This characteristic limits the current iteration's applicability in ultra-low-latency real-time scenarios.

Furthermore, the robustness of our reward mechanism relies on the quality of the off-the-shelf NLI model (e.g., RoBERTa-large-mnli). These auxiliary models are generally effective for standard domains. However, they may overlook subtle contradictions in highly specialized fields, such as law or medicine. Addressing this requires domain-specific fine-tuning. Finally, our MMKP selector aims to maximize information density. It operates on the assumption that redundancy within retrieved groups is semantically uniform. In cases of high semantic complexity, rigorous filtering might inadvertently discard complementary minority opinions.

Ethical Considerations

Our work aims to enhance the reliability of Large Language Models. We focus on reducing hallucinations through constrained context selection and logical verification. By grounding generation in traceable evidence, we mitigate risks associated with disseminating factually incorrect information. However, we acknowledge the environmental impact of the test-time compute paradigm. The iterative nature of Monte Carlo Tree Search increases GPU utilization compared to standard decoding methods. Consequently, this results in higher energy consumption. Future work should focus on optimizing the planner's sample efficiency to reduce this computational footprint.

Additionally, as a Retrieval-Augmented Generation system, our model's outputs are constrained by the retrieved corpora. The results reflect the quality and potential biases of the source documents. Our NLI guidance penalizes logical contradictions. However, it does not verify the intrinsic factual accuracy of the retrieved text. If the knowledge base contains biased or toxic content, the system may reproduce these issues. It is important to note that faithfulness in this context refers to adherence to the retrieved context. It does not necessarily equate to absolute objective truth.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 52578347).

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, and 26 others. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *Preprint*, arXiv:2204.01691.

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

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of thoughts: Solving elaborate problems with large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *Preprint*, arXiv:2303.12712.

Ruijun Chen, Jiehao Liang, Shiping Gao, Fanqi Wan, and Xiaojun Quan. 2024. Self-evolution fine-tuning for policy optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4120–4137.

Rong Fu, Yang Li, Zeyu Zhang, Jiekai Wu, Yaohua Liu, Shuaishuai Cao, Yangchen Zeng, Yuhang Zhang, Xiaojing Du, Chuang Zhao, Kangning Cui, and Simon Fong. 2026a. Neurosymactive: Differentiable neural-symbolic reasoning with active exploration for knowledge graph question answering. *Preprint*, arXiv:2602.15353.

Rong Fu, Chunlei Meng, Youjin Wang, Haoyu Zhao, Jiaxuan Lu, Kun Liu, JiaBao Dou, and Simon James Fong. 2026b. Neuropareto: Calibrated acquisition for costly many-goal search in vast parameter spaces. *Preprint*, arXiv:2602.03901.

Rong Fu, Yemin Wang, Tianxiang Xu, Yongtai Liu, Weizhi Tang, Wangyu Wu, Xiaowen Ma, and Simon Fong. 2026c. S-path-rag: Semantic-aware shortest-path retrieval augmented generation for multi-hop knowledge graph question answering. *arXiv preprint arXiv:2603.23512*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Xueren Ge, Sahil Murtaza, Anthony Cortez, and Homa Alemzadeh. 2026. Expert-guided prompting and retrieval-augmented generation for emergency medical service question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 30798–30806.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2025. Evoprompt: Connecting llms with evolutionary algorithms yields powerful prompt optimizers. *Preprint*, arXiv:2309.08532.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173, Singapore. Association for Computational Linguistics.

Maolin He, Rena Gao, Mike Conway, and Brian E. Chapman. 2024a. Query pipeline optimization for cancer patient question answering systems. *Preprint*, arXiv:2412.14751.

Zhenyu He, Zexuan Zhong, Tianle Cai, Jason Lee, and Di He. 2024b. REST: Retrieval-based speculative decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1582–1595, Mexico City, Mexico. Association for Computational Linguistics.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *Preprint*, arXiv:2011.01060.

Wentao Hu, Wengyu Zhang, Yiyang Jiang, Chen Jason Zhang, Xiaoyong Wei, and Li Qing. 2025. Removal of hallucination on hallucination: Debate-augmented RAG. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15839–15853, Vienna, Austria. Association for Computational Linguistics.

Fanding Huang, Guanbo Huang, Xiao Fan, Yi He, Xiao Liang, Xiao Chen, Qinting Jiang, Faisal Nadeem Khan, Jingyan Jiang, and Zhi Wang. 2026. Semantic-space exploration and exploitation in rlvr for llm reasoning. *Preprint*, arXiv:2509.23808.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Dongming Jiang, Yi Li, Guanpeng Li, and Bingzhe Li. 2026a. Magma: A multi-graph based agentic memory architecture for ai agents. *Preprint*, arXiv:2601.03236.

Dongming Jiang, Yi Li, Songtao Wei, Jinxin Yang, Ayushi Kishore, Alysa Zhao, Dingyi Kang, Xu Hu, Feng Chen, Qiannan Li, and Bingzhe Li. 2026b. Anatomy of agentic memory: Taxonomy and empirical analysis of evaluation and system limitations. *Preprint*, arXiv:2602.19320.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 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, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Bo Li, Tian Tian, Zhenghua Xu, Hao Cheng, Shikun Zhang, and Wei Ye. 2026a. Modeling uncertainty

trends for timely retrieval in dynamic RAG. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 31527–31535. AAAI Press.

Bo Li, Mingda Wang, Gexiang Fang, Shikun Zhang, and Wei Ye. 2026b. Retrieval as generation: A unified framework with self-triggered information planning. *Preprint*, arXiv:2604.11407.

Guocong Li, Jinjian Zhang, Ping Wang, Dongnan Liu, Tian Liang, Qiuyi Qi, Hao Huang, Siyan Guo, Mutian Bao, Wei Zhou, Linjian Mo, Hongxia Xu, and Jian Wu. 2026c. Mol: Adaptive mixture-of-length reasoning for efficient question answering with context. In *The Fourteenth International Conference on Learning Representations*.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. Halueval: A large-scale hallucination evaluation benchmark for large language models. *Preprint*, arXiv:2305.11747.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023b. Api-bank: A comprehensive benchmark for tool-augmented llms. *Preprint*, arXiv:2304.08244.

Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. 2023. Llm+p: Empowering large language models with optimal planning proficiency. *Preprint*, arXiv:2304.11477.

Jieyi Long. 2023. Large language model guided tree-of-thought. *Preprint*, arXiv:2305.08291.

Guangtao Lyu, Xinyi Cheng, Chenghao Xu, Qi Liu, Muli Yang, Fen Fang, Huilin Chen, Jiexi Yan, Xu Yang, and Cheng Deng. 2025. Revealing perception and generation dynamics in lvlms: Mitigating hallucinations via validated dominance correction. *arXiv preprint arXiv:2512.18813*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023a. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023b. Zero-shot listwise document reranking with a large language model. *Preprint*, arXiv:2305.02156.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. 2024. Gorilla: Large language model connected with massive apis. In *Advances in Neural Information Processing Systems*, volume 37, pages 126544–126565. Curran Associates, Inc.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *Preprint*, arXiv:2309.15088.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *Preprint*, arXiv:2307.16789.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar

Fawzi, and 1 others. 2024. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pages 68539–68551. Curran Associates, Inc.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 8634–8652. Curran Associates, Inc.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2024. Is chatgpt good at search? investigating large language models as re-ranking agents. *Preprint*, arXiv:2304.09542.

Yuxi Sun, Aoqi Zuo, Wei Gao, and Jing Ma. 2025. Causalabstain: Enhancing multilingual llms with causal reasoning for trustworthy abstention. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14060–14076.

Yuxi Sun, Aoqi Zuo, Haotian Xie, Wei Gao, Mingming Gong, and Jing Ma. 2026. Fact-e: Causality-inspired evaluation for trustworthy chain-of-thought reasoning. *Preprint*, arXiv:2604.10693.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *Preprint*, arXiv:2401.15391.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

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

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, volume 36, pages 74952–74965. Curran Associates, Inc.

Chengbing Wang, Yang Zhang, Wenjie Wang, Xiaoyan Zhao, Fuli Feng, Xiangnan He, and Tat-Seng Chua. 2026a. Think-while-generating: On-the-fly reasoning for personalized long-form generation. *Preprint*, arXiv:2512.06690.

Chengbing Wang, Wuqiang Zheng, Yang Zhang, Fengbin Zhu, Junyi Cheng, Yi Xie, Wenjie Wang, and Fuli Feng. 2026b. Perm: Psychology-grounded empathetic reward modeling for large language models. *Preprint*, arXiv:2601.10532.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Xudong Wang, Chaoning Zhang, Qigan Sun, Zhenzhen Huang, Chang Lu, Sheng Zheng, Zeyu Ma, Caiyan Qin, Yang Yang, and Hengtao Shen. 2026c. Transforming external knowledge into triplets for enhanced retrieval in rag of llms. *Preprint*, arXiv:2604.12610.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023a. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023b. Learning to filter context for retrieval-augmented generation. *Preprint*, arXiv:2311.08377.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Wang Xi, Quan Shi, Zenghui Ding, Jianqing Gao, and Xianjun Yang. 2025. Hallucination as a computational boundary: A hierarchy of inevitability and the oracle escape. *Preprint*, arXiv:2508.07334.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 641–649, New

York, NY, USA. Association for Computing Machinery.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *Preprint*, arXiv:2310.04408.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024a. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.

Tianyu Yang, Yiyang Nan, Lisen Dai, Zhenwen Liang, Yapeng Tian, and Xiangliang Zhang. 2024b. Sasr-net: Source-aware semantic representation network for enhancing audio-visual question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15894–15904, Miami, Florida, USA. Association for Computational Linguistics.

Tianyu Yang, Sihong Wu, Yilun Zhao, Zhenwen Liang, Lisen Dai, Chen Zhao, Minhao Cheng, Arman Cohan, and Xiangliang Zhang. 2026. Deconstructing multimodal mathematical reasoning: Towards a unified perception-alignment-reasoning paradigm. *Preprint*, arXiv:2603.08291.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William 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, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Haoran Ye, Jiarui Wang, Zhiguang Cao, Federico Berto, Chuanbo Hua, Haeyeon Kim, Jinkyoo Park, and Guojie Song. 2024. Reevo: Large language models as hyper-heuristics with reflective evolution. In *Advances in Neural Information Processing Systems*, volume 37, pages 43571–43608. Curran Associates, Inc.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488. Curran Associates, Inc.

Fangyuan Zhang, Zhengjun Huang, Yingli Zhou, Qintian Guo, Zhixun Li, Wensheng Luo, Di Jiang, Yixiang Fang, and Xiaofang Zhou. 2025a. Erarag: Efficient and incremental retrieval augmented generation for growing corpora. *arXiv preprint arXiv:2506.20963*.

Qianchi Zhang, Hainan Zhang, Liang Pang, Hongwei Zheng, and Zhiming Zheng. 2026. Stable-rag: Mitigating retrieval-permutation-induced hallucinations in retrieval-augmented generation. *Preprint*, arXiv:2601.02993.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2025b. Siren's song in the ai ocean: A survey on hallucination in large language models. *Computational Linguistics*, pages 1–46.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2024. Language agent tree search unifies reasoning acting and planning in language models. *Preprint*, arXiv:2310.04406.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. *Preprint*, arXiv:2205.10625.

Xiaofan Zhou, Huy Nguyen, Bo Yu, Chenxi Liu, and Lu Cheng. 2026a. Adaptive stopping for multi-turn llm reasoning. *Preprint*, arXiv:2604.01413.

Yixi Zhou, Fan Zhang, Yu Chen, Haipeng Zhang, Preslav Nakov, and Zhuohan Xie. 2026b. Fincards: Card-based analyst reranking for financial document question answering. *Preprint*, arXiv:2601.06992.

Appendices

Within this supplementary material, we elaborate on the following aspects:

- Appendix A: Dataset Details
- Appendix B: Implementation Details and Hyperparameters
- Appendix C: Theoretical Proofs and Analysis
- Appendix D: Detailed Experimental Results
- Appendix E: Sensitivity Analysis Results
- Appendix F: LLM Prompts
- Appendix G: Qualitative Analysis Case Study

A Dataset Details

We evaluate our approach on six representative datasets covering Simple QA, Multi-Hop QA, and Multi-Document QA scenarios. The statistics of the evaluation datasets are summarized in Table 8.

Simple QA. We first consider single-hop tasks that require retrieving facts from open-domain sources.

- **Natural Questions (NQ):** An open-domain QA dataset. In our experiments, we use a subset of 1,000 queries and retrieve from a corpus of 9,633 passages.
- **PopQA:** This dataset focuses on long-tail entities which often require precise knowledge retrieval. We evaluate on 1,000 queries with a retrieval pool of 8,676 passages.

Multi-Hop QA. These datasets require the model to perform reasoning across multiple documents (typically 2–4 hops) to derive the answer.

- **MuSiQue:** A challenging multi-hop dataset. Our evaluation involves 1,000 queries and the largest passage pool in our setup (11,656 passages), requiring complex reasoning chains.
- **2WikiMultiHopQA (2Wiki):** Based on Wikipedia, requiring 2–4 hops. We use 1,000 queries and 6,119 passages.
- **HotpotQA:** We utilize the distractor setting to test the model's ability to filter irrelevant information. The setup includes 1,000 queries and 9,811 passages.

Multi-Doc QA. Finally, we evaluate robustness against noise in a retrieval-augmented generation setting.

- **MultiHop-RAG:** Designed to test noise robustness. Unlike the standard QA sets, this evaluation uses a larger set of 2,556 queries over 609 specific passages to measure the F1 score and retrieval accuracy.

B Implementation Details and Hyperparameters

B.1 General Implementation Details

We implemented our framework using PyTorch and the Hugging Face Transformers library. All experiments were conducted on a cluster of NVIDIA A100 (80GB) GPUs. For the retrieval backbone, we utilized a hybrid approach combining dense and sparse retrieval. We employed `BAAI/bge-small-en-v1.5` as the dense embedding model and `scikit-learn`'s `TfidfVectorizer` (ngram range 1-2, max features 200k) for sparse retrieval. The results were fused using Reciprocal Rank Fusion (RRF) with $k = 60$, alongside a centroid-based query expansion strategy where the query embedding is refined using the mean of the top-5 dense retrieval results. For the generator, we utilized `Qwen2.5-7B-Instruct` as the backbone Large Language Model (LLM), loaded in `bfloat16` precision. The NLI verification signal was derived from `roberta-large-mnli`.

B.2 Baseline Implementation Details

To ensure a rigorous comparison, we aligned the base models and resource constraints across all baselines where applicable:

- **Standard RAG (Naive):** We follow the standard "Retrieve-then-Generate" paradigm. We retrieve the top-k ($k = 5$) chunks using the same hybrid retriever (BGE + TF-IDF) described above and feed them directly into the `Qwen2.5-7B-Instruct` model. No advanced selection or re-ranking is applied.
- **RAG + MMR:** We implemented Maximal Marginal Relevance (MMR) to re-rank the retrieval candidates. We set the diversity hyperparameter $\lambda = 0.6$, iteratively selecting documents that maximize a linear combination of query relevance and novelty relative to

Category	Dataset	Type	Avg. Hops	# Queries	# Passages	Metric
Simple QA	NQ	Open-domain	1	1,000	9,633	EM / F1
	PopQA	Long-tail Entity	1	1,000	8,676	EM / F1
Multi-Hop QA	MuSiQue	Multi-hop	2.40	1,000	11,656	EM / F1
	2WikiMultiHopQA	Multi-hop	2.42	1,000	6,119	EM / F1
	HotpotQA	Multi-hop	2	1,000	9,811	EM / F1
Multi-Doc QA	MultiHop-RAG	Noise Robustness	2.70	2,556	609	EM / F1

Table 8: Statistics of the evaluation datasets. The "Queries" and "Passages" columns denote the specific counts used in our experimental evaluation.

the already selected set, until the token budget (1500 tokens) is exhausted.

- **Self-RAG:** We utilized the official `selfrag/selfrag-llama2-7b` checkpoint. To maintain fairness in information access, we restricted the retrieved context size provided to Self-RAG to be approximately equivalent to our MMKP token budget (~ 5 chunks or 1500 tokens).

B.3 Our Method: Configuration and Logic

B.3.1 MMKP Context Selector

The MMKP selector transforms the retrieval list into a constrained optimization problem. We first grouped retrieval candidates using a greedy clustering algorithm based on cosine similarity thresholds ($\tau_{sim} = 0.82$). For each candidate document d_{ij} in group G_i, we calculated:

- **Value (v_{ij}):** A weighted sum of relevance and diversity: $v_{ij} = \alpha \cdot \text{Score}_{\text{fusion}}(q, d_{ij}) + \beta \cdot (1 - \text{Sim}(d_{ij}, \mu_{G_i}))$, where μ_{G_i} is the group centroid.
- **Costs (w_{ij}):** A 2-dimensional cost vector containing the token length (L_{token}) and a semantic redundancy penalty (C_{red}). The redundancy penalty is calculated as the mean similarity to other group members, scaled by a factor of 100.

We solved this NP-hard problem using a Dynamic Programming approach with Pareto pruning to remove dominated states (states with higher costs and lower value), ensuring tractability.

B.3.2 NLI-Guided MCTS Generator

We implemented the generator as a Monte Carlo Tree Search (MCTS) process that explores the reasoning space at test time.

- **Node Expansion:** At each step, the model chooses between two action types: *Answer* (generate a response hypothesis) or *Augment* (retrieve additional evidence using the current query).
- **Reward Function:** We employed `roberta-large-mnli` to compute a faithfulness reward. The reward R is defined as $R = w_e \cdot P(entail) + w_n \cdot P(neutral) + w_c \cdot P(contradict)$. We set a severe penalty for contradictions ($w_c = -2.0$) to aggressively prune hallucinatory paths.
- **Search Strategy:** We used the UCT (Upper Confidence Bound for Trees) algorithm with an exploration constant $c_{ucb} = 1.4$ to balance exploration of new reasoning paths and exploitation of high-reward trajectories.

B.4 Hyperparameters

The specific hyperparameters for the MMKP selector and MCTS generator used in our main experiments are detailed in Table 9 and Table 10.

Parameter	Value
Token Budget (C_{token})	1500
Redundancy Budget (C_{red})	120
Relevance Weight (α)	0.7
Diversity Weight (β)	0.3
Similarity Threshold (τ_{sim})	0.82
Redundancy Cost Scale	100.0

Table 9: Hyperparameters for the MMKP Context Selector. Weights were tuned on the HotpotQA validation set.

Parameter	Value
Generator (LLM)	
Model	Qwen2.5-7B-Instruct
Temperature	0.7
Top-p	0.9
Max New Tokens	256
MCTS Search	
Simulations (N)	24
Branching Factor (k)	3
Max Search Depth	3
Exploration Constant (c_{ucb})	1.4
Reward Function (NLI)	
Entailment Weight (w_e)	1.0
Neutral Weight (w_n)	-0.2
Contradiction Weight (w_c)	-2.0

Table 10: Hyperparameters for NLI-Guided MCTS. The generator uses these settings during the test-time search phase.

C Theoretical Proofs and Analysis

C.1 Supplementary Definitions for MMKP

C.1.1 Multidimensional Cost Vectors

Unlike the standard Knapsack problem (single weight), RAG systems face multiple resource constraints. We define the cost vector $\mathbf{w}_{ij} \in \mathbb{R}^K$ for each item. We model $K = 2$ dimensions:

1. Token Consumption ($k = 1$): $w_{ij}^{(1)} = \text{Len}(d_{ij})$.
2. Redundancy Penalty ($k = 2$): Derived from the intra-group centroid.

$$w_{ij}^{(2)} = \lambda_{\text{red}} \cdot \left(\frac{1}{|G_i| - 1} \sum_{d \in G_i \setminus \{d_{ij}\}} \cos\big(\Phi(d_{ij}), \Phi(d)\big) \right) \tag{3}$$

This cost dimension penalizes items that are too central or generic within their cluster, preferring unique information, scaled by λ_{red}.

C.1.2 Utility Function

The utility v_{ij} balances query relevance and global diversity:

$$v_{ij} = \alpha \cdot \mathcal{F}_{fusion}(q, d_{ij}) + \beta \cdot (1 - \text{Sim}(d_{ij}, \mu_{G_i})) \tag{4}$$

where $\mathcal{F}_{fusion}$ incorporates both dense and sparse (TF-IDF) retrieval scores (via Reciprocal Rank Fusion), and μ_{G_i} is the centroid of group G_i.

C.2 Computational Complexity of MMKP

Theorem 1 (NP-hardness). *The RAG Document Selection problem formulated as MMKP (Eq. 1) is NP-hard.*

Proof. We perform a reduction from the classic 0/1 Knapsack Problem, which is known to be NP-hard. Let a standard Knapsack instance be defined by n items with values v_i, weights w_i, and capacity W. We construct a special instance of our RAG-MMKP as follows:

1. Create $m = n$ groups.
2. Each group G_i contains exactly two items: $\{d_{i,1}, d_{i,0}\}$.
3. Item $d_{i,1}$ (representing "taking" item i) has value v_i and weight vector $\mathbf{w}_{i,1} = [w_i, 0, \ldots, 0]$.
4. Item $d_{i,0}$ (representing "leaving" item i) has value 0 and weight vector $\mathbf{0}$.
5. Set the MMKP capacity vector $\mathbf{C} = [W, \infty, \ldots, \infty]$.

The constraint $\sum_j x_{ij} \leq 1$ in MMKP forces the selection of exactly one item per group (either the real item or the dummy zero-cost item), which is mathematically equivalent to the binary choice $x_i \in \{0, 1\}$ in the standard Knapsack problem. Since 0/1 Knapsack is NP-hard, and it is a special case of RAG-MMKP, RAG-MMKP is NP-hard. ■

C.3 FPTAS for Core MMKP Variant

We analyze the single-dimensional variant ($D = 1$) where each group has size 1 (equivalent to standard Knapsack). We prove the existence of a Fully Polynomial-Time Approximation Scheme (FPTAS).

Theorem 2 (FPTAS). *For any $\epsilon > 0$, there exists an algorithm that returns a solution S such that $V(S) \geq (1 - \epsilon)\, OPT$ and runs in time polynomial in n and $1/\epsilon$.*

Algorithm Construction:

1. Let $P = \max_i v_i$. Given error tolerance $\epsilon > 0$, define scaling factor $K = \frac{\epsilon P}{n}$.
2. Define scaled values $v'_i = \lfloor \frac{v_i}{K} \rfloor$.
3. Solve the problem using DP on values v'_i. The recurrence is: $DP[k, v] = \min(DP[k-1, v], w_k + DP[k-1, v - v'_k])$. The max possible scaled value is $V'_{max} \approx n \cdot \frac{P}{K} = \frac{n^2}{\epsilon}$.

Algorithm 1 FPTAS for 0/1 Knapsack Problem

procedure FPTAS-KNAPSACK(v, w, C, ϵ)
  Let n be the number of items.
  $P \leftarrow \max_{i=1}^{n} v_i$
  $K \leftarrow \frac{\epsilon P}{n}$
  For $i = 1, \ldots, n : v'_i \leftarrow \lfloor v_i / K \rfloor$
  $V'_{max} \leftarrow \sum_{i=1}^{n} v'_i$
  Initialize DP table T of size $V'_{max} + 1$ with $T[0] = 0$ and $T[p] = \infty$ for $p > 0$.
  for $i = 1, \ldots, n$ **do**
    for $p = V'_{max}, \ldots, v'_i$ **do**
      $T[p] \leftarrow \min(T[p], w_i + T[p - v'_i])$
  Find the largest p^* such that $T[p^*] \leq C$.
  Reconstruct the set of items corresponding to $T[p^*]$ and return it.

The algorithm is detailed in Algorithm 1.

Proof. Let S^* be the optimbual set and S be the set returned by our algorithm; we prove the approximation guarantee. By definition of floor: $Kv'_i \leq v_i < K(v'_i + 1)$. The optimal value $OPT = \sum_{i \in S^*} v_i$. Our algorithm finds S that optimizes the scaled values, so $\sum_{i \in S} v'_i \geq \sum_{i \in S^*} v'_i$.

Considering the lower bound of our solution $V(S)$, we have:

$$\begin{aligned} V(S) = \sum_{i \in S} v_i &\geq \sum_{i \in S} K v'_i = K \sum_{i \in S} v'_i \\ &\geq K \sum_{i \in S^*} v'_i \\ &> K \sum_{i \in S^*} \left(\frac{v_i}{K} - 1 \right) \\ &= \sum_{i \in S^*} v_i - \sum_{i \in S^*} K \\ &= OPT - nK \end{aligned} \tag{5}$$

Substituting $K = \frac{\epsilon P}{n}$, we get $V(S) > OPT - \epsilon P$. Since $OPT \geq P$, we have $V(S) \geq OPT - \epsilon OPT = (1 - \epsilon) OPT$.

Time Complexity: The DP table size is $O(n \cdot V'_{max}) = O\left(n \cdot \frac{n^2}{\epsilon}\right) = O\left(\frac{n^3}{\epsilon}\right)$. This is polynomial in n and $1/\epsilon$, satisfying the definition of FPTAS. ■

C.4 Heuristic DP Details and Pareto Pruning

Pareto-Pruned Dynamic Programming. For the practical implementation where $D = 2$, we utilize a Dynamic Programming approach with state pruning. Let $DP[i]$ be the set of achievable states after considering group G_i. Each state is a tuple $(\mathbf{c}, v)$, representing the accumulated cost vector and value. To avoid state explosion, we prune dominated states. A state $A = (\mathbf{c}_A, v_A)$ dominates $B = (\mathbf{c}_B, v_B)$ if and only if:

$$v_A \geq v_B \quad \text{and} \quad \forall k : c_A^{(k)} \leq c_B^{(k)} \tag{6}$$

At each step i, we compute

$$DP[i] = \text{PF}\Bigg(\Big\{ (\mathbf{c} + \mathbf{w}_{ij}, v + v_{ij}) \mid (\mathbf{c}, v) \in DP[i-1],\ d_{ij} \in G_i \Big\} \Bigg) \tag{7}$$

This reduces the average complexity to polynomial time in practice.The algorithm is detailed in Algorithm 2.

Algorithm 2 MMKP with Pareto-Pruned DP

Input: Groups $\mathcal{G}$, Budgets $\mathbf{C}$
$DP \leftarrow \{(0, 0) : 0.0\}$ ▷ Map (cost_tokens, cost_red) → value
for each group $G_i \in \mathcal{G}$ **do**
  $DP_{new} \leftarrow DP$.copy()
  for each item $d_{ij} \in G_i$ **do**
    for each state $(\mathbf{c}, v) \in DP$ **do**
      $\mathbf{c}' \leftarrow \mathbf{c} + \mathbf{w}_{ij}$
      $v' \leftarrow v + v_{ij}$
      if $\mathbf{c}' \preceq \mathbf{C}$ **then**
        Update DP_{new} with $(\mathbf{c}', v')$
  $DP \leftarrow$ PruneDominated(DP_{new})
return $\max_v DP$

The function `PruneDominated` removes any state $(\mathbf{c}_B, v_B)$ if there exists $(\mathbf{c}_A, v_A)$ such that $\mathbf{c}_A \leq \mathbf{c}_B$ AND $v_A \geq v_B$. This ensures we only track the Pareto frontier.

C.5 Detailed MCTS Search Procedure

We employ a variant of the Upper Confidence Bound for Trees (UCT) & PUCT-style selection. The search proceeds in four steps for N_{sim} simulations:

1. **Selection:** Starting from root s_0, we recursively select child nodes satisfying:

$$a^* = \underset{a \in \mathcal{A}(s)}{\text{argmax}} \Bigg(Q(s, a) + c_{puct} \cdot P(a|s) \frac{\sqrt{N(s)}}{1 + N(s, a)} \Bigg) \tag{8}$$

where $Q(s,a)$ is the estimated value, $N(s)$ is the visit count, and $P(a|s)$ is the prior from the policy (LLM).

2. **Expansion:** If node s is not a terminal state, we expand it by sampling k candidate answers (Generative Actions) and optionally m retrieval queries (Augmentative Actions).

3. **Simulation (Rollout):** From the expanded node, we perform a rollout using the base policy π_θ to generate a complete answer y_{final}.

4. **Backpropagation:** We compute the reward $R(y_{final})$ using Eq. 2 and update the Q-values along the trajectory:

$$Q(s,a) \leftarrow \frac{N(s,a) \cdot Q(s,a) + R}{N(s,a) + 1} \quad (9)$$

C.6 Convergence Analysis of NLI-Guided MCTS

We provide a theoretical justification for the use of UCT in the space of logical consistency.

Theorem 3 (MCTS Consistency). *As the number of simulations $N \to \infty$, the probability of selecting the optimal answer y^* (defined as the answer maximizing the NLI reward) approaches 1.*

Proof. The RAG generation process is modeled as a finite-horizon MDP. The NLI reward $R(y) \in [-2, 1]$ is bounded. The UCT selection rule is:

$$\bar{X}_j + 2C_p\sqrt{\frac{2\ln n}{n_j}} \quad (10)$$

According to the Kocsis and Szepesvári (2006) theorem, UCT is consistent in finite-horizon domains. The regret R_n after n steps grows as $O(\ln n)$. Specifically for our NLI-guided generation:

1. **Exploration:** The $w_{con} = -2.0$ penalty in our reward function (Eq. 2) acts as a soft pruning mechanism. Branches containing contradictions yield low Q-values.

2. **Exploitation:** UCT exponentially allocates samples to branches with high entailment scores ($w_{ent} = 1.0$).

Therefore, provided the NLI model Θ is an approximate oracle of truth, the search policy π_{MCTS} converges to the sentence sequence y that maximizes logical entailment with the evidence set $\mathcal{E}$. ■

D Detailed Experimental Results

We further investigate the source of our performance gains by analyzing the impact of token constraints and reasoning complexity.

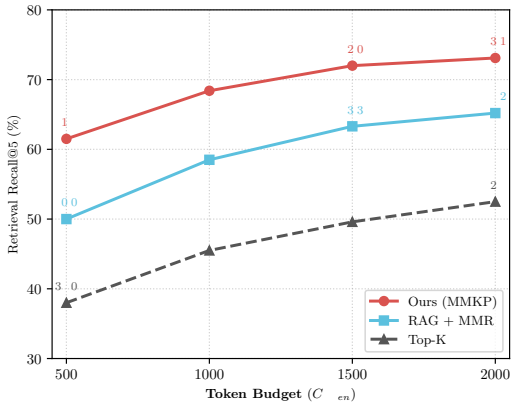

Figure 3: Impact of token budget on retrieval performance. This figure compares the Retrieval Recall@5 of our MMKP method against RAG+MMR and Top-K baselines across varying token budgets (C_{token}). Our MMKP (red line) consistently outperforms the baselines, demonstrating superior robustness.

Token Budget Robustness. Figure 3 demonstrates the retrieval recall across varying token constraints. While traditional Top-K selection suffers a severe performance drop to 38.0% when the budget is restricted to 500 tokens, our MMKP selector exhibits superior resilience. It maintains a high recall of 61.5% in this strict setting, outperforming Top-K by a margin of 23.5% and the RAG+MMR baseline by 11.5%. This substantial gap validates MMKP's ability to maximize information density when context space is scarce.

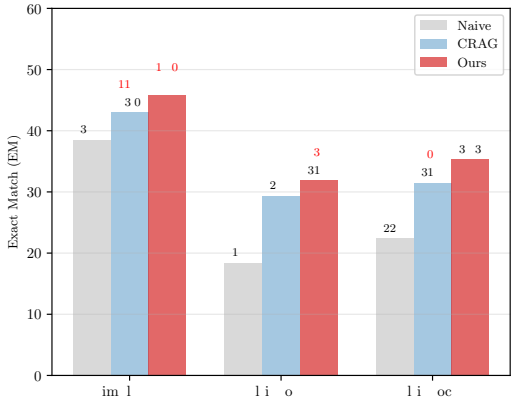

Figure 4: Complexity Analysis. Our method shows significant scaling advantages on harder queries.

Reasoning on Hard Queries. Figure 4 presents

a performance decomposition based on query complexity. While our method attains a 19.0% improvement over the Naive baseline on Simple QA, the performance gains are substantially amplified in complex reasoning scenarios. Specifically, on Multi-Hop QA, our approach demonstrates superior scaling by achieving a 73.4% relative gain over the baseline and exceeding the CRAG model by 2.5 absolute points. Similarly, for Multi-Doc QA, our model records a 57.6% improvement over the baseline. This result significantly widens the margin compared to CRAG, thereby validating the efficacy of our self-correction mechanism in handling long-context and multi-step reasoning tasks.

E Sensitivity Analysis Results

In this appendix, we provide the detailed experimental results supporting the sensitivity analysis discussed in Section 6. We evaluate the impact of the redundancy budget on retrieval recall and the effect of MCTS simulation parameters on generation performance.

E.1 Redundancy Budget in MMKP

Figure 5 illustrates the relationship between the redundancy budget (C_{red}) and retrieval performance. The redundancy score is calculated based on maximum semantic similarity between selected passages.

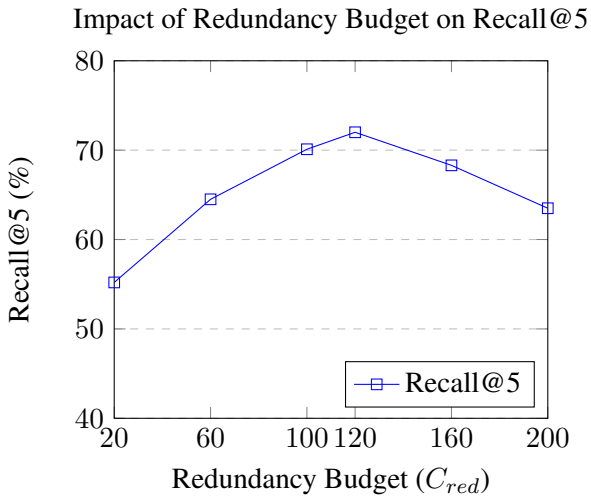

Figure 5: Sensitivity of Recall@5 to the redundancy budget (C_{red}). A budget of approx. 120 provides the optimal trade-off, allowing sufficient diversity without excluding relevant evidence.

E.2 MCTS Simulation Parameters

Table 11 details the generation performance (F1 Score) across different configurations of the MCTS planner. We vary the number of simulations (N) and the maximum search depth (D).

Simulations (N)	Depth (D)	F1 Score	Faithfulness (AP)
1 (Greedy)	1	36.1	0.52
8	2	40.5	0.68
16	3	43.2	0.79
24	**3**	**45.8**	**0.85**
32	4	46.0	0.86
64	5	46.1	0.87

Table 11: Effect of MCTS hyperparameters on performance. N denotes the number of simulations, and D denotes the search depth. The chosen configuration ($N = 24, D = 3$) is highlighted.

As observed, increasing N beyond 24 yields diminishing returns in F1 score. Similarly, extending the depth beyond 3 provides marginal gains in faithfulness but complicates the reasoning trace unnecessarily for most tasks.

F LLM Prompts

We employ two distinct sets of prompts for our framework: one for the core Retrieval-Augmented Generation (RAG) tasks and another for the symbolic planning module. Figure 6 illustrates the prompts used for the NLI-Guided Generator, and Figure 7 shows the prompts for the Symbolic Plan Generator.

G Qualitative Analysis Case Studies

To investigate the mechanisms by which Self-Correcting RAG rectifies errors, we analyze specific failure cases of the baseline compared to our framework. We present two distinct scenarios: one focusing on context truncation due to redundancy, and another on attribute comparison amidst high-information noise.

G.1 Case Study 1: Temporal Comparison with Context Truncation

In this first scenario, we examine a temporal comparison query regarding the founding dates of two magazines. The baseline fails due to context truncation of the second entity, leading to a hallucinated date. In contrast, our method rectifies this via (1) MMKP-based context de-duplication and (2) MCTS-guided verification.

INSTRUCTION:

You are a rigorous multi-hop QA assistant. You must **answer strictly based on the provided document snippets**. Do not introduce external knowledge, subjective guesses, or information not mentioned in the documents. Please think deeply and answer according to the following steps:

1. **Decompose the Question:** Break down the core query points and the logical link of multi-hop reasoning (e.g., "Precondition → Intermediate Inference → Final Conclusion").
2. **Locate Relevant Documents:** Examine candidate document snippets one by one, mark content directly/indirectly related to the core query, and record the corresponding `doc_id`. Exclude irrelevant documents.
3. **Integrate Information:** If relevant information is fragmented, logically connect it based on the document context. If there is no direct answer, provide a **speculative conclusion consistent with the document context** (do not return "unknown").
4. **Verify Rationality:** Confirm that the answer originates entirely from the documents, contains no external information, and does not contradict the document content.
5. **Output Result:** Provide a concise answer, the corresponding evidence `doc_ids`, and explain the deep thinking process.

INPUT TEMPLATE:

Question: {question}

Candidate Document Snippets (Listed by `doc_id`, potentially from different articles; content truncated):
{context}

Please strictly output in the following JSON format. The "reasoning" field must detail the deep thinking process (including question decomposition, document location, information integration, and rationality verification, at least 3 sentences):

```
{
  "answer": "<Concise answer based on docs / Speculative conclusion>",
  "evidence_doc_ids": ["<doc_id1>", "<doc_id2>", "..."],
  "reasoning": "<Detailed thinking process...>"
}
```

Figure 6: The prompt template used for the NLI-Guided Generator. It enforces strict adherence to retrieved context and requires explicit reasoning steps to support the MCTS verification process.

Figure 8 illustrates a detailed distinct failure mode analysis. In the **Baseline (Left)**, the dense retriever retrieves three documents with high cosine similarity (> 0.88) to the entity "The American Conservative". However, due to the limited context window (Top-3 constraints), these semantically redundant chunks crowd out the essential document containing the founding date of the second entity, "The Weekly Standard". Consequently, the CoT reasoner, lacking specific evidence, falls back on parametric memory, hallucinating an incorrect date (1950s) based on a broad "Cold War era" bias.

In contrast, our **Self-Correcting Framework (Right)** intervenes at two stages:

1. **Context Construction:** The MMKP module clusters retrieved chunks by semantic intent. It identifies that Doc 1 and Doc 2 convey identical information (Cluster A) and discards the lower-ranked duplicate, effectively reserving token budget for the diverse Cluster B (Doc 3).
2. **Reasoning Verification:** The MCTS planner expands the search space. When the model initially generates a hallucinated date (Branch π_1), the NLI verifier detects a low entailment score (0.12) against the retrieved context, assigning a negative reward ($r = -1.0$). This prompts the planner to backtrack and explore Branch π_2, which successfully extracts the correct date verified by high entailment (0.98), leading to the correct temporal comparison.

G.2 Case Study 2: Attribute Comparison under Information Noise

We further analyze a multi-hop reasoning scenario involving corporate history, which typically re-

INSTRUCTION:

You are a plan generator that outputs only JSON. You will not provide thinking processes.
Task: Generate an executable PLAN (sequence of steps) for the question using the following primitives:

- `FIND(by: entity|number|year|text, value)`
- `FILTER(by: year|number, cond: =, >, <, >=, <= with numeric)`
- `SELECT(keyword)` *# Filter sentences by keyword*
- `AGG(type: COUNT|MAX|MIN, target?: year|number)`
- `COMPARE(mode: MAX|MIN)` *# Find sentence maximizing target in candidates*
- `ANSWER(template: string with {value})`

Only return JSON: `{"steps":[ ... ]}`, do not output any other content.

CONSTRAINTS & REQUIREMENTS:

1) If the question involves numerical values/dates/comparisons, please use `FIND + FILTER/AGG/COMPARE`.
2) You must end with `ANSWER`. Use `{value}` as a placeholder in the template. If there is no numerical value, you can treat `{value}` as the final answer directly.
3) Do not include explanations or extra fields.

INPUT:

Question: {question}

Only output JSON.

Figure 7: The prompt template used for the Symbolic Plan Generator. This prompt restricts the LLM to output a structured execution plan using predefined API primitives.

quires bridging entity relationships and precise attribute comparison. The query asks: *"Who had a longer tenure as CEO of the company that acquired DeepMind: Eric Schmidt or Larry Page?"*

This query presents a specific challenge: "DeepMind" is a keyword heavily associated with recent AI breakthroughs, creating a "distractor dense" retrieval environment.

As shown in Figure 9, the **Baseline** suffers from *topic drift*. The dense retriever prioritizes documents about DeepMind's technical achievements (AlphaGo, AlphaZero) due to high semantic similarity with the query entity. These redundant documents occupy the limited context window, crowding out the essential biographical data regarding the CEOs of the parent company (Google). Consequently, the model relies on a "Founder Heuristic"—assuming the founder (Larry Page) served the longest—resulting in a hallucinated conclusion.

In contrast, our **Self-Correcting RAG** demonstrates robustness:

1. **MMKP Filtering:** The Context Selector groups the "AI Achievement" documents into a single semantic cluster. It recognizes that selecting multiple documents from this cluster offers diminishing returns. It effectively prunes the redundancy (similar to the mechanism described in Section G.1), freeing up token budget to retrieve the specific tenure dates for both Schmidt and Page.
2. **MCTS Verification:** The generator initially attempts a heuristic guess (Page $>$ Schmidt). However, the NLI-guided verifier checks this against the retrieved evidence (Schmidt: 10 years vs. Page: 4 years) and penalizes the contradiction ($Q = -1.0$). The planner then backtracks to perform the correct arithmetic comparison, producing a faithful result.

INPUT QUERY & COMPLEXITY ANALYSIS

Query: *"Which magazine was founded first, 'The American Conservative' (TAC) or 'The Weekly Standard' (TWS)?"*

Reasoning Requirement: 1. **Multi-hop Retrieval:** Must find $Date(TAC)$ AND $Date(TWS)$. 2. **Temporal Comparison:** Compute $\min(Date_{TAC}, Date_{TWS})$. *Hardness: High (Ambiguity in entity dates)*

Baseline: Standard RAG (Failure Trace)

STEP 1: DENSE RETRIEVAL (TOP-K)
Retriever matches query embedding to chunks. Limit: $K = 3$.

- **[Rank 1] Doc A:** "TAC is a bi-monthly journal... founded in **2002** by Pat Buchanan..."
  → Sim: 0.89 (High relevance to 'TAC')
- **[Rank 2] Doc B:** "TAC (Website) offers opinion pieces... launched in 2002..."
  → Sim: 0.88 (Redundant Semantic Cluster)
- **[Rank 3] Doc C:** "Pat Buchanan co-founded TAC to promote anti-war views..."
  → Sim: 0.86 (Redundant Semantic Cluster)

CRITICAL ERROR: *Context window full. The document containing "The Weekly Standard (1995)" (Rank 4, Sim 0.84) is* ***truncated****.*

STEP 2: COT REASONING (HALLUCINATION)
LLM Input: Doc A, Doc B, Doc C.

- **Thought 1:** "Founding date of TAC is 2002 (from Doc A)."
- **Thought 2:** "Searching for TWS date... Not in context."
- **Thought 3 (Parametric Memory):** "TWS is a famous neo-con magazine. It feels older, associated with the 90s or Cold War. Let's assume **1950s**."
- **Comparsion:** $1950s < 2002$.

Final Prediction:
The Weekly Standard (approx. 1950s)
✗ *Factually Incorrect Reason*

Ours: Self-Correcting RAG (Success Trace)

PHASE 1: MMKP CONTEXT SELECTOR
Goal: Maximize Diversity / Minimize Redundancy.

- **Cluster 1 (Topic: TAC):** {Doc A, Doc B, Doc C}
- **Cluster 2 (Topic: TWS):** {Doc D (Rank 4)}
- **Action:** Select representative from Cluster 1 (Doc A) → **Discard Doc B & C.** → **Retrieve Doc D.**

Optimized Context: {Doc A (2002), Doc D ("TWS... founded Sep 17, 1995")}

PHASE 2: MCTS PLANNER REASONING
Policy: $\pi(a|s)$*, Reward: NLI(Context, Hypothesis).*
Node 0: Root State (Question)

↪ **Branch 1: Greedy Generation (Hallucination)**
Hypothesis: "TWS founded in 1955."
→ **Verification (NLI):** Context (Doc D: 1995) vs Hypothesis (1955).
→ **Result: Contradiction** (0.99).
→ **Reward** $Q(s, a)$**: -1.0** *(Prune path)*

↪ **Branch 2: Guided Extraction (Correct)**
Hypothesis: "TWS founded in 1995."
→ **Verification (NLI):** Context (Doc D: 1995) vs Hypothesis (1995).
→ **Result: Entailment** (0.98).
→ **Reward** $Q(s, a)$**: +1.0** *(Proceed)*

Final Prediction:
The Weekly Standard (1995)
✓ *Factually Grounded*

Figure 8: **Comprehensive Trace of Failure vs. Correction. Left:** The Baseline fails due to *information crowding*. High-similarity redundant documents about Entity A fill the context window, cutting off Entity B. The model then hallucinates to fill the gap. **Right:** Our approach employs **MMKP** (Maximal Marginal Relevance based Knapsack Problem) to filter semantic duplicates, ensuring both entities are present. Subsequently, the **MCTS** (Monte Carlo Tree Search) planner explores reasoning paths. It actively penalizes the hallucinated branch (Branch 1) via NLI verification and rewards the factually consistent branch (Branch 2).

INPUT QUERY & COMPLEXITY ANALYSIS

Query: *"Who had a longer tenure as CEO of the company that acquired DeepMind: Eric Schmidt or Larry Page?"*

Reasoning Requirement: 1. **Entity Bridging:** Identify Parent Company (DeepMind → Google). 2. **Attribute Retrieval:** Find CEO Tenure(Schmidt) AND CEO Tenure(Page). 3. **Numerical Comparison:** Calculate Δt. *Hardness: High (Distractor Noise)*

Baseline: Standard RAG (Failure Trace)

STEP 1: DENSE RETRIEVAL (TOP-K)
Retriever matches "DeepMind" to high-ranking news chunks.

- **[Rank 1] Doc A:** "Google acquires AI startup DeepMind for $500M in 2014..."
  → Sim: 0.91 (Relevant Bridge)
- **[Rank 2] Doc B:** "DeepMind's AlphaGo defeats Lee Sedol in historic match..."
  → Sim: 0.89 (Distractor - High Semantic Overlap)
- **[Rank 3] Doc C:** "Demis Hassabis discusses DeepMind's AlphaZero algorithm..."
  → Sim: 0.87 (Distractor - Redundant Topic)

CRITICAL ERROR: *Distractor Crowding.* High-scoring documents about DeepMind's AI achievements fill the context. *Evidence for Schmidt's tenure is pushed to Rank 7 and truncated.*

STEP 2: COT REASONING (HALLUCINATION)
LLM Input: Doc A, Doc B, Doc C.

- **Thought 1:** "DeepMind was acquired by Google (Doc A)."
- **Thought 2:** "Need tenures for Eric Schmidt and Larry Page."
- **Thought 3 (Missing Info):** "Context discusses AlphaGo, not CEO dates."
- **Thought 4 (Heuristic Bias):** "Larry Page is the co-founder of Google. Founders usually serve the longest terms. Eric Schmidt was hired later."

Final Prediction:
Larry Page (Founder Heuristic)
✗ *Factually Incorrect (Page: ≈4 yrs vs Schmidt: 10 yrs)*

Ours: Self-Correcting RAG (Success Trace)

PHASE 1: MMKP CONTEXT SELECTOR
Goal: Penalize Semantic Redundancy (C_{red}).

- **Cluster 1 (DeepMind News):** {Doc B (AlphaGo), Doc C (AlphaZero)}
- **Cluster 2 (Corporate Structure):** {Doc A (Acquisition), Doc D (Schmidt Tenure), Doc E (Page Tenure)}
- **Action:** Detect high redundancy in Cluster 1. **Prune Doc B & C.** Allocate budget to Cluster 2.

Optimized Context: {Doc A, Doc D ("Schmidt CEO 2001-2011"), Doc E ("Page CEO 2011-2015")}

PHASE 2: MCTS PLANNER REASONING
Node 0: Root State (Question)

↪ **Branch 1: Heuristic Assumption (Fail)**
Hypothesis: "Larry Page served longer."
→ **Verification (NLI):** Context (Schmidt: 10 yrs, Page: 4 yrs) vs Hypothesis.
→ **Result: Contradiction** (0.95).
→ **Reward: -1.0** *(Prune)*

↪ **Branch 2: Calculation (Success)**
Step 1: Schmidt: $2011 - 2001 = 10$ years.
Step 2: Page: $2015 - 2011 = 4$ years.
Hypothesis: "Eric Schmidt (10 years) > Larry Page."
→ **Verification (NLI): Entailment** (0.99).
→ **Reward: +1.0** *(Proceed)*

Final Prediction:
Eric Schmidt (10 years)
✓ *Factually Grounded Calculation*

Figure 9: **Analysis of Distractor Filtering and Numerical Verification. Left:** The Baseline fails due to *Distractor Crowding*. Popular documents about DeepMind's AI achievements (AlphaGo) overwhelm the context window, displacing the necessary CEO tenure dates. **Right:** The **MMKP Selector** identifies the "AI Achievement" documents as semantically redundant and removes them. This preserves space for documents containing specific tenure dates. The **MCTS Planner** then rejects the heuristic bias ("Founders serve longer") via NLI verification, ensuring the final answer is derived from arithmetic comparison of the retrieved dates.