ADADR-RAG: ADAPTIVE AGENTIC RAG GUIDED BY DIVERSITY AND RELIABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Early Retrieval-Augmented Generation (RAG) approaches typically adopt a static paradigm where models passively incorporate externally retrieved knowledge to generate answers. In contrast, Agentic RAG reframes question answering as a dynamic interaction between the model and its environment characterized by an alternating cycle of reasoning and search engine invocation. However, existing studies in Agentic RAG suffer from two primary limitations. First, they retrieve a fixed number of knowledge snippets at each interaction step regardless of information sufficiency. Second, they predominantly focus on final answer accuracy while neglecting the diversity and reliability of the iterative interaction trajectories. To address these challenges, we propose AdaDR-RAG, an Adaptive Diversity and Reliability-guided Agentic RAG framework. First, an adaptive search controller dynamically modulates retrieval depth with an exploration-exploitation balancing mechanism, ensuring sufficient evidence accumulation while preventing degenerate query loops and context pollution. Second, a trajectory clustering strategy extracts cluster-level rewards to jointly optimize for diversity and reliability, preventing policy collapse via sparsity-aware incentives and ensuring factual consistency through intra-group consensus to stabilize multi-step reasoning. Experimental results demonstrate that our method outperforms state-of-the-art baselines on widely-used benchmarks for both general and multi-hop question answering.

1 INTRODUCTION

Retrieval-Augmented Generation (RAG) has emerged as a cornerstone paradigm for enhancing Large Language Models (LLMs) with external, up-to-date knowledge. Traditional RAG systems (Lewis et al., 2020; Guu et al., 2020; Ke et al., 2024; Chen et al., 2025) typically operate under static workflows that passively incorporate retrieved knowledge without adapting to evolving query demands. While effective for straightforward queries, this passive paradigm struggles with complex, multi-hop questions that demand iterative reasoning and dynamic information seeking. To address this limitation, Agentic RAG (Trivedi et al., 2023; Li et al., 2025; Jin et al., 2025) reframes question answering as an interactive decision-making process. As illustrated in Figure 1, the model autonomously alternates between internal deliberation (`<think>`), dynamic query formulation (`<search>`), and evidence synthesis (`<information>`), progressively decomposing complex questions into subgoals. This active information-seeking paradigm enables the agent to iteratively accumulate and validate external context before generating a final answer, using new evidence to guide the next reasoning or search decision in a manner that mirrors the iterative nature of human problem-solving.

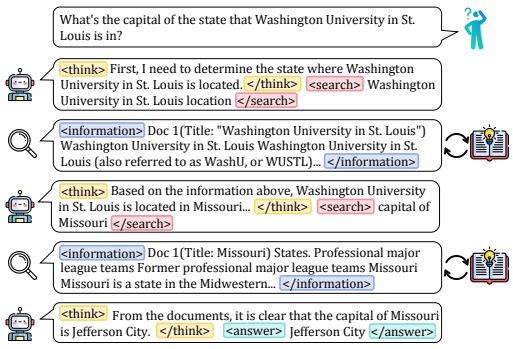


Figure 1: Illustrative multi-hop QA workflow demonstrating the iterative reasoning and dynamic retrieval paradigm of Agentic RAG.

Despite promising progress, existing Agentic RAG approaches (Li et al., 2025; Wu et al., 2025; Zhang et al., 2026) face two critical bottlenecks. First, they rely on rigid retrieval schedules that

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

fetch a fixed number of documents per interaction step, irrespective of actual information sufficiency. This rigid retrieval schedule frequently mismatches variable query complexity, leading to either context pollution from redundant retrieval or truncated reasoning. Second, current optimization paradigms predominantly target terminal answer correctness.

Although recent works (Wu et al., 2025; Zhang et al., 2026; Xia et al., 2026) incorporate process supervision, they typically either depend on external verifiers or focus narrowly on step-level accuracy. These studies failed to account for exploration diversity and factual reliability. Consequently, agents tend to prematurely converge on dominant but suboptimal paths, exhibiting brittle behavior when confronted with complex or ambiguous queries.

To address these challenges, we propose **AdaDR-RAG** (Adaptive Diversity and Reliability-guided Agentic RAG), a framework that jointly optimizes adaptive retrieval and trajectory-level reasoning quality. Specifically, we introduce *Query-Aware Adaptive Agentic Search*, which dynamically modulates retrieval depth via implicit query comparison. By progressively fetching cached document batches only when information is insufficient, it ensures thorough evidence accumulation while eliminating redundant context pollution. An integrated exploration-exploitation balancing mechanism further stabilizes this process by regulating query switching and deepening frequencies, effectively preventing degenerative search loops while preserving flexible reasoning control. Second, to move beyond correctness-centric optimization, we develop *Diverse and Reliable Agentic Policy Optimization*. By grouping rollouts via structural query alignment, it derives complementary reward signals that promote exploration diversity to prevent policy collapse and enforce factual reliability through intra-cluster answer consensus. This formulation jointly balances exploratory breadth with factual consistency, enabling the agent to autonomously discover robust retrieval strategies without handcrafted rules or external critics. Our main contributions are as follows:

- We propose a sufficiency-guided adaptive search controller that dynamically decides whether to advance reasoning or deepen search by progressively paging cached results for the same query, thereby ensuring query-level information completeness while maintaining optimal context utilization.
- We introduce a cluster-based dual-reward framework that assigns diversity and reliability signals based on trajectory group statistics, leveraging inverse cluster size and group accuracy consensus to jointly optimize exploration breadth and answer reliability.
- Extensive experiments on widely-used general and multi-hop QA benchmarks demonstrate that AdaDR-RAG consistently outperforms state-of-the-art baselines, achieving superior accuracy with more stable interaction trajectories.

2 Related Work

2.1 Static Workflow-based RAG

Early RAG systems operate under a static retrieve-then-generate paradigm, fetching a fixed set of documents before answer synthesis (Lewis et al., 2020; Guu et al., 2020). To enhance retrieval quality within fixed workflows, recent approaches have integrated systematic optimization signals. BGM (Ke et al., 2024) aligns retriever outputs with generator preferences by optimizing document ranking through direct preference feedback, while MMOA-RAG (Chen et al., 2025) coordinates retriever and generator policies via multi-agent reinforcement learning to improve single-round retrieval relevance. However, their static retrieval workflows lack the flexibility to dynamically adapt to intermediate reasoning states, limiting their effectiveness on complex multi-hop queries.

2.2 Agentic RAG

To overcome static pipeline rigidity, Agentic RAG reframes question answering as an interactive decision-making process, enabling iterative information seeking through dynamic reasoning and retrieval (Yao et al., 2023; Qin et al., 2024). Early approaches rely on prompt engineering (Trivedi et al., 2023), recently extended by frameworks like Search-o1 (Li et al., 2025) that use specialized instructions to interleave autonomous retrieval and knowledge refinement in long-horizon reasoning. However, these prompt-based methods lack systematic optimization signals. To address this, reinforcement learning approaches optimize search policies via policy gradients, often leveraging critic-

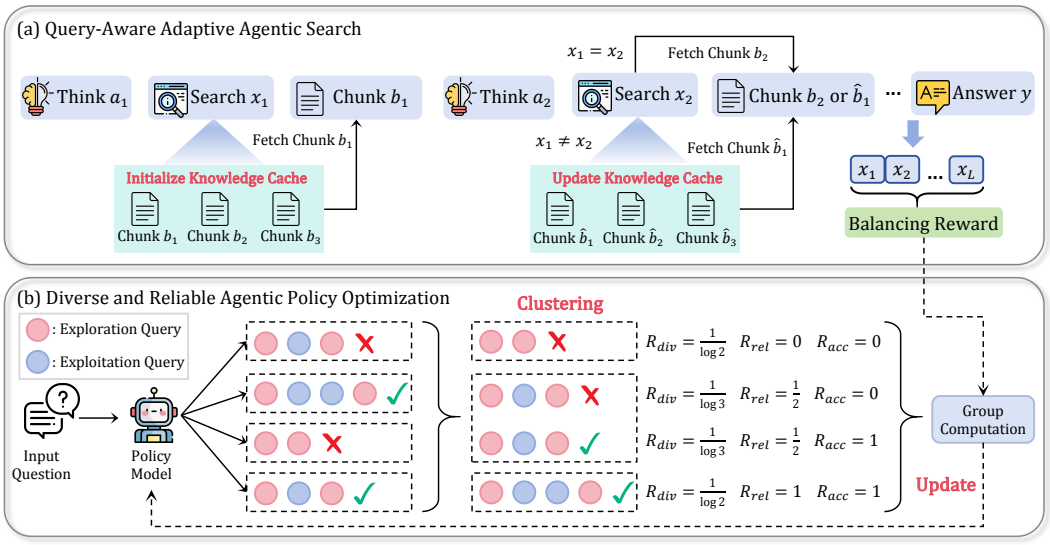


Figure 2: Overview of the **AdaDR-RAG** framework. **(a) Query-Aware Adaptive Agentic Search:** Retrieval depth is dynamically modulated via implicit query comparison, adaptively balancing exploration and exploitation to mitigate information insufficiency and redundancy. **(b) Diverse and Reliable Agentic Policy Optimization:** Sampled trajectories are clustered by structural similarity. Diversity and reliability signals are derived from cluster statistics and synthesized to steer policy refinement through GRPO.

free algorithms like GRPO (Shao et al., 2024). Specifically, Search-R1 (Jin et al., 2025) leverages outcome-based rewards to guide interleaved reasoning processes. To address retrieval inefficiencies, HiPRAG (Wu et al., 2025) introduces hierarchical process rewards that explicitly penalize both excessive and insufficient search steps. TreePS-RAG (Zhang et al., 2026) employs Monte Carlo tree estimation to derive fine-grained, step-level advantages for more precise decision-making. Furthermore, Search-P1 (Xia et al., 2026) enhances training robustness by extracting learning signals from suboptimal trajectories via path-centric soft scoring. However, existing methods still rely on rigid retrieval schedules and correctness-centric objectives, limiting adaptive depth and trajectory diversity. AdaDR-RAG overcomes this via implicit query-guided continuation and a structured optimization framework that jointly balances exploration diversity and factual reliability.

3 Methodology

3.1 Task Formulation

We formulate Agentic Retrieval-Augmented Generation (Agentic RAG) as a sequential decision-making process. Given a query q, the agent interacts with an external search environment over a horizon of T steps. At each step t ($t \leq T$), the policy π_θ generates a reasoning trace a_t and a search query x_t. The environment returns a retrieved document batch $\mathcal{B}_t$, and the state updates as $s_{t+1} = \text{Concat}(s_t, a_t, x_t, \mathcal{B}_t)$. A complete rollout yields a trajectory $\tau = \{(a_1, x_1, \mathcal{B}_1), \ldots, (a_L, x_L, \mathcal{B}_L)\}$ with $L \leq T$. Following recent agentic RL work (Jin et al., 2025; Xia et al., 2026), we generate a group of G trajectories (rollouts) $\{\tau^{(1)}, \ldots, \tau^{(G)}\}$ per query at each training step and employ GRPO (Shao et al., 2024) as the training framework.

3.2 Method Overview

As illustrated in Figure 2, the architecture integrates two synergistic modules. Specifically, Section 3.3 introduces a query-aware adaptive agentic search mechanism that adjusts retrieval pagination via implicit query comparison, ensuring information sufficiency while mitigating context redundancy. Building upon this, Section 3.4 and Section 3.5 present a unified Diverse and Reliable Agentic Policy Optimization framework. This module first employs a trajectory clustering strategy

to partition sampled rollouts into structural groups based on sequential query alignment, and subsequently derives structured exploration incentives and group-level consensus metrics from these clusters. These combined signals are then integrated into the GRPO objective to update the policy.

3.3 Query-Aware Adaptive Agentic Search

Fixed-step retrieval often introduces irrelevant context or prematurely truncates reasoning. AdaDR-RAG employs an implicit control signal derived from the agent's query generation behavior. Specifically, this interactive behavior is governed by a structured system prompt (detailed in Appendix J) that explicitly instructs the model to autonomously decide whether to repeat a query for iterative deepening, reformulate it to adjust search scope, or terminate with a final answer. During rollout, we maintain a query-result cache $\mathcal{K}$ that maps each unique query to its top-M retrieved document batches: $\mathcal{K}[x_t] = [b_1, b_2, \ldots, b_M]$, where b_m denotes the m-th retrieved chunk. Let x_t denote the query generated at step t. We compare x_t with the preceding query x_{t-1} to dynamically adjust retrieval depth:

$$k_t = \begin{cases} k_{t-1} + 1, & \text{if } x_t = x_{t-1} \text{ and } x_{t-1} \neq \emptyset \\ 1, & \text{otherwise} \end{cases} \tag{1}$$

where k_t denotes the dynamic retrieval depth (cache page pointer) for query x_t. It explicitly dictates which document batch $\mathcal{B}_t = b_{k_t}$ is injected into the context at step t. When k_t increases, the agent performs iterative deepening on a fixed query. When k_t resets to 1, the agent pivots to a novel subgoal, triggering a fresh search and caching process. The interaction terminates upon the generation of a final answer, or when k_t reaches the predefined maximum depth $K_{\max}$.

To prevent the agent from exploiting this mechanism through degenerate repetition loops or excessive query thrashing, we introduce a trajectory-level exploration-exploitation balancing reward. For each rollout $\tau^{(g)}$, we extract the query trajectory $X^{(g)} = (x_1^{(g)}, \ldots, x_{L_g}^{(g)})$ of length L_g and compute the consecutive repetition count $n_{\text{rep}} = \sum_{t=2}^{L_g} \mathbb{I}(x_t^{(g)} = x_{t-1}^{(g)})$. The balancing reward is:

$$R_{\text{bal}} = \begin{cases} \lambda_{\text{bal}}, & L_g \leq L_{min} \\ \lambda_{\text{bal}} \cdot \exp\left(-\frac{D(r)^2}{2\sigma^2}\right), & L_g > L_{min} \end{cases} \tag{2}$$

where $r = \frac{n_{\text{rep}}}{L_g - 1}$ denotes the consecutive repetition ratio and $D(r) = \max(0, r - r_{\text{high}}) + \max(0, r_{\text{low}} - r)$ measures deviation from the tolerance band $[r_{\text{low}}, r_{\text{high}}]$. The reward stays constant at λ_{bal} within the tolerance band but decays exponentially when r falls outside this range. This formulation balances query deepening with semantic switching while explicitly penalizing degenerate search loops. To prevent unfair penalization from extreme ratios that inherently violate the tolerance band, we introduce a minimum length threshold $L_{\min}$ and directly assign the baseline reward λ_{bal} for $L_g \leq L_{\min}$. The scale parameter σ modulates the penalty steepness to ensure smooth gradient updates for stable optimization.

3.4 Query Trajectory Clustering

To distinguish robust retrieval strategies from stochastic exploration and ensure stable reward signals across training iterations, we maintain a per-query historical trajectory buffer H_q. For each query q in the training batch, H_q is persistently updated across iterations by accumulating the query trajectories $\{X^{(1)}, \ldots, X^{(G)}\}$ extracted from the current rollouts: $H_q \leftarrow H_q \cup \{X^{(1)}, \ldots, X^{(G)}\}$. By aggregating historical trajectories, we reduce sensitivity to fluctuations in the current rollout batch and maintain a persistent pool of search patterns. This pool preserves diverse retrieval strategies explored at different stages of training. To extract coherent retrieval strategies from H_q, we formulate a trajectory clustering mechanism that partitions these query sequences based on their consistency.

Within the buffer H_q, we measure the semantic alignment between any two query trajectories $X^{(i)}$ and $X^{(j)}$ using dense sentence embeddings. Specifically, each query step is encoded via a pre-trained sentence encoder (`all-MiniLM-L6-v2`), and the trajectory-level similarity is computed as the average step-wise cosine similarity:

$$S_{ij} = \mathbb{I}(L_i = L_j) \cdot \frac{1}{L_i} \sum_{l=1}^{L_i} \cos\left(\mathbf{e}_l^{(i)}, \mathbf{e}_l^{(j)}\right), \tag{3}$$

where $\mathbf{e}_l^{(i)}$ denotes the normalized embedding of the l-th query in trajectory $X^{(i)}$. The indicator $\mathbb{I}(L_i = L_j)$ ensures that only trajectories with identical exploration depths are compared, preserving step-wise correspondence. By averaging similarities across synchronized steps, this metric captures intent-level consistency while naturally tolerating paraphrases or synonyms.

Based on the pairwise similarity matrix computed over H_q, we apply agglomerative clustering to partition the buffered query trajectories into disjoint structural groups $\{\mathcal{C}_1, \ldots, \mathcal{C}_I\}$. Starting from each trajectory as a singleton cluster, this bottom-up procedure iteratively merges the two most similar groups based on average linkage until a predefined similarity threshold δ is met. This topology-aware grouping transforms isolated query sequences into coherent strategy clusters, enabling fine-grained differentiation of search patterns for subsequent policy optimization.

3.5 Diverse and Reliable Agentic Policy Optimization

Leveraging the clustered topology, we formulate a structured guidance strategy that explicitly balances exploration breadth with factual consistency. For each trajectory $X^{(i)}$ assigned to cluster $\mathcal{C}$, the policy objective jointly incorporates a density-based exploration incentive, a consensus-driven reliability measure, and a terminal accuracy term aligned with ground-truth verification.

Diversity Reward. Without explicit exploration regularization, agentic policies tend to converge toward high-probability query formulations, restricting exploration breadth and causing the agent to overlook alternative evidence paths. This tendency is particularly problematic in multi-hop reasoning, where critical information is often distributed across diverse retrieval routes. To counteract this convergence bias, we introduce a diversity reward that inversely scales with trajectory density:

$$R_{\text{div}}^{(i)} = \frac{1}{\log(1 + |\mathcal{C}|)}. \tag{4}$$

Trajectories in smaller, underexplored clusters receive higher incentives. This formulation provides a bounded, smoothly decaying incentive that scales with cluster sparsity, ensuring sustained exploration without destabilizing the primary accuracy-driven optimization objective.

Reliability Reward. While diversity promotes broad exploration, unconstrained trajectories often suffer from factual instability. To ground exploration in factual consistency, we introduce the reliability reward, which evaluates retrieval robustness via intra-cluster consensus. Rather than relying on isolated terminal signals, it directly aggregates intra-cluster answer correctness. By rewarding query sequences that consistently yield accurate results across reasoning variations, this mechanism suppresses reward variance and provides a stable optimization signal. Formally, the reliability reward is computed as:

$$R_{\text{rel}}^{(i)} = \frac{1}{|\mathcal{C}|} \sum_{X^{(j)} \in \mathcal{C}} \mathbb{I}(y^{(j)} = y_q^*). \tag{5}$$

where $y^{(j)}$ is the answer to the j-th sequence and y_q^* is the ground truth. This group-wise aggregation transforms sparse terminal feedback into a continuous reliability signal. It inherently filters out trajectories whose correctness stems from stochastic variations or weak factual grounding, ensuring that policy optimization is guided by factually grounded and consistent retrieval strategies. This stable learning signal reduces policy oscillation and promotes convergence in multi-hop reasoning.

Accuracy Reward. To maintain direct alignment with the ground-truth objective, we retain a standard terminal signal:

$$R_{\text{acc}}^{(i)} = \mathbb{I}(y^{(i)} = y_q^*), \tag{6}$$

ensuring that structural exploration does not diverge from factual correctness.

The unified trajectory reward is formulated as:

$$R^{(i)} = R_{\text{div}}^{(i)} + R_{\text{rel}}^{(i)} + R_{\text{acc}}^{(i)} + R_{\text{bal}}^{(i)}, \tag{7}$$

This design directly operationalizes the diversity-reliability trade-off, enabling the agent to autonomously discover novel yet dependable retrieval strategies without relying on handcrafted rules or external critics.

The policy π_θ is optimized via GRPO. For query q, the group rewards $\{R^{(1)}, \ldots, R^{(G)}\}$ are normalized to obtain group-relative advantages $A_i = \frac{R^{(i)} - \mu_R}{\sigma_R + \gamma}$, where γ prevents division by zero, and μ and σ are the mean and standard deviation of the group rewards, respectively:

$$\mu_R = \frac{1}{G}\sum_{j=1}^{G} R^{(j)}, \quad \sigma_R = \sqrt{\frac{1}{G}\sum_{j=1}^{G}\left(R^{(j)} - \mu_R\right)^2}. \tag{8}$$

The surrogate objective maximizes the clipped policy likelihood while constraining deviation from a frozen reference policy π_{ref} via KL divergence:

$$\begin{aligned}\mathcal{L}(\theta) = \mathbb{E}_{q,\tau^{(i)}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\min\Big(r_i(\theta)A_i, \\ \text{clip}(r_i(\theta), 1-\epsilon, 1+\epsilon)A_i\Big)\Bigg] - \beta_{\text{KL}}\mathcal{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}),\end{aligned} \tag{9}$$

where $r_i(\theta) = \frac{\pi_\theta(\tau^{(i)}|q)}{\pi_{\theta_{\text{old}}}(\tau^{(i)}|q)}$. To focus optimization on the agent's decision-making process and prevent gradient interference from static context, we apply a loss mask to all observation tokens (i.e., retrieved passages). Consequently, gradient updates are computed exclusively over the model-generated outputs. This targeted optimization strategy stabilizes training dynamics and enables the policy to jointly refine its multi-step reasoning capabilities, query formulation strategies, and adaptive retrieval depth. The complete training iteration is summarized in Algorithm 1 (Appendix L).

4 Experiments

4.1 Experimental Settings

Datasets and Metrics. Following previous research (Jin et al., 2025; Zhang et al., 2026), we evaluated the performance of AdaDR-RAG on seven benchmarks (statistical information is in Appendix A): (1) General QA: NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023); (2) Multi-hop QA: HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), Musique (Trivedi et al., 2022), and Bamboogle (Press et al., 2023). We use Exact Match (EM) as the evaluation metric to measure performance.

Baselines. We compare AdaDR-RAG with the following five types of baseline methods: (1) Reasoning without retrieval (Direct): method using direct prompting; (2) Standard RAG: performing single-round retrieval to obtain external knowledge for generation; (3) RL-based static workflow RAG (BGM (Ke et al., 2024), and MMOA-RAG (Chen et al., 2025)): optimizing single-round retrieval through feedback from retrieval signals; (4) Prompt-based agent RAG (IRCoT (Trivedi et al., 2023) and Search-o1 (Li et al., 2025)): interacting with the external environment through prompting methods; (5) RL-based agent RAG (Search-R1 (Jin et al., 2025), HiPRAG (Wu et al., 2025), Search-P1 (Xia et al., 2026) and TreePS-RAG (Zhang et al., 2026)): optimizing the model's environment interaction ability through reinforcement learning training methods.

Implementation Details. We conducted experiments using `Qwen2.5-3B-Instruct` and `Qwen2.5-7B-Instruct` (Qwen et al., 2025) as backbone models. For retrieval, we used the 2018 Wikipedia dump as the knowledge base and E5 (Wang et al., 2024) as the retriever. The lower and upper bounds of the tolerance band are set to 0.3 and 0.7, respectively. Please refer to Appendix B for more implementation details.

4.2 Main Results

Table 1 reports the comprehensive evaluation results. AdaDR-RAG achieves state-of-the-art performance on most benchmarks. Compared to static RAG and prompt-based agentic methods, our approach yields substantial improvements, confirming the necessity of dynamic interaction and systematic policy optimization. Against recent RL-based agentic baselines (Search-R1, HiPRAG, TreePS-RAG), which increasingly leverage process supervision yet limit reward signals to step-level correctness, AdaDR-RAG achieves a +3.6% average gain over the strongest baseline by explicitly integrating diversity and reliability guidance into policy optimization.

Table 1: Main results on seven public QA benchmarks. Best results are in **bold**, second best are underlined. AG and RL respectively indicate whether they are based on agentic search or involve RL-based training.

Method	AG	RL	General QA			Multi-Hop QA				Avg.
			NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	
Qwen2.5-3B-Instruct										
Direct	✗	✗	10.6	28.8	10.8	14.9	24.4	2.0	2.4	13.4
RAG	✗	✗	34.8	54.4	38.7	25.5	22.6	4.7	8.0	27.0
BGM	✗	✓	42.6	57.0	39.5	32.1	30.8	11.7	20.8	33.5
MMOA-RAG	✗	✓	47.0	57.7	39.7	34.7	34.8	13.9	24.0	36.0
IRCoT	✓	✗	11.1	31.2	20.0	16.4	17.1	6.7	24.0	18.1
Search-o1	✓	✗	23.8	47.2	26.2	22.1	21.8	5.4	32.0	25.5
Search-R1	✓	✓	39.7	56.5	39.1	33.1	31.0	12.4	23.2	33.6
HiPRAG	✓	✓	43.0	59.8	42.0	36.0	40.5	10.8	24.0	36.6
Search-P1	✓	✓	53.0	**74.5**	47.9	36.2	36.6	13.3	28.8	41.5
TreePS-RAG	✓	✓	47.6	63.6	46.0	46.2	45.6	21.5	44.0	44.9
AdaDR-RAG	✓	✓	**55.5**	71.2	**51.3**	**48.2**	**49.0**	**22.8**	**47.2**	**49.3**
Qwen2.5-7B-Instruct										
Direct	✗	✗	13.4	40.8	14.0	18.3	25.0	3.1	12.0	18.1
RAG	✗	✗	34.9	58.5	39.2	29.9	23.5	5.8	20.8	30.4
BGM	✗	✓	46.9	66.3	44.0	37.4	33.9	15.6	38.4	40.4
MMOA-RAG	✗	✓	47.2	63.3	46.3	40.5	36.1	17.9	40.0	41.6
IRCoT	✓	✗	22.4	47.8	30.1	13.3	14.9	7.2	22.4	23.9
Search-o1	✓	✗	15.1	44.3	13.1	18.7	17.6	5.8	29.6	20.6
Search-R1	✓	✓	42.9	62.3	42.7	38.6	34.6	16.2	40.0	39.6
HiPRAG	✓	✓	46.5	65.8	45.8	42.0	46.1	14.0	40.0	42.9
Search-P1	✓	✓	56.6	**78.6**	47.5	42.9	39.8	21.8	44.0	47.3
TreePS-RAG	✓	✓	52.1	67.6	48.5	50.7	48.2	23.9	52.0	49.0
AdaDR-RAG	✓	✓	**58.3**	74.6	**50.1**	**52.5**	**51.6**	**25.7**	**55.2**	**52.6**

The performance advantage is most pronounced on challenging multi-hop datasets like 2Wiki and Bamboogle, where questions demand iterative evidence accumulation and exploratory search. In these scenarios, our cluster-guided reward mechanism effectively encourages the agent to bypass dominant but suboptimal paths. On general QA benchmarks (NQ, TriviaQA, PopQA), AdaDR-RAG maintains strong accuracy by dynamically balancing information sufficiency against contextual redundancy, thereby avoiding the context pollution caused by over-retrieval or the reasoning failures caused by under-retrieval. Furthermore, the framework demonstrates robust scaling: it retains a consistent lead over all baselines with both Qwen2.5-3B and 7B, indicating that our diversity and reliability optimization effectively stabilizes agentic interaction and compensates for variations in model capacity.

4.3 Ablation Study

Building on the main results in Table 1, which demonstrate AdaDR-RAG's superiority over state-of-the-art baselines, we conduct a systematic ablation study on `Qwen2.5-7B-Instruct` to quantify the contribution of each component. The results are summarized in Table 2.

Adaptive Agentic Search. Adaptive continuation adjusts retrieval depth to the evidence available for each query. Removing the adaptive continuation mechanism (`w/o Ada`) forces the model to retrieve a fixed number of passages ($k = 3$) per search action. As shown in Table 2, this leads to a consistent performance drop across all benchmarks. This confirms that query-level adaptive paging is crucial for maintaining context sufficiency while minimizing redundant information. Furthermore, ablating the trajectory-level balancing reward (`w/o` R_{bal}) results in comparable degradation. Without the tolerance-band regularization, the policy loses control over intra-trajectory search depth, manifesting in two failure modes: excessive repetition of identical queries (redundant paging loops) or overly frequent pivoting to unrelated sub-goals (shallow exploration thrashing). This demonstrates that R_{bal} is essential for maintaining a stable exploitation-exploration trade-off during retrieval, effectively preventing degenerate trajectories while preserving query-level adaptivity.

Table 2: Ablation results for AdaDR-RAG components on four QA benchmarks with Qwen2.5-7B-Instruct.

Methods	NQ	PopQA	HotpotQA	2Wiki
AdaDR-RAG	**58.3**	**50.1**	**52.5**	**51.6**
w/o Ada	56.1	47.3	49.8	48.6
w/o R_{bal}	57.4	49.3	51.3	50.1
w/o R_{div}	57.8	49.6	50.8	49.2
w/o R_{rel}	56.9	48.5	50.1	48.7
whole-clustering	57.5	49.2	50.6	49.3
step-exact	56.7	48.7	49.6	48.8
rel-max	57.6	49.0	51.3	49.3

Query Trajectory Clustering. We compare our embedding-based trajectory clustering against two variants to examine step alignment and semantic matching. Concatenating the full query sequence into a single string (whole-clustering) underperforms because global string alignment is highly sensitive to insertion or deletion shifts. A changed query can disrupt the alignment of later steps, even when the overall search intent is similar. This global approach therefore fails to capture synchronized step-wise correspondence. Using step-wise exact string matching instead of dense embeddings (step-exact) degrades significantly because differently phrased queries can seek the same evidence. The dense embedding process requires only a lightweight sentence encoder, introducing negligible latency during each clustering step while generalizing to paraphrased search strategies without losing the alignment between successive retrieval steps across query trajectories. This preserves comparisons of semantically related search paths.

Diversity Reward. Agent-based retrieval systems inherently gravitate toward high-probability query templates, frequently overlooking alternative evidence paths. To mitigate this convergence bias, R_{div} explicitly rewards underrepresented strategies by assigning higher incentives to trajectories within smaller clusters. Ablating this component (w/o R_{div}) causes a notable accuracy decline, especially on complex reasoning tasks (e.g., HotpotQA and 2Wiki). Without diversity signals, the agent rapidly collapses into a narrow set of familiar strategies, missing critical multi-hop evidence that requires traversing non-dominant search routes.

Reliability Reward. To enforce epistemic rigor within the search space, R_{rel} evaluates group-level consensus by averaging answer correctness across trajectories in the same cluster, effectively smoothing reward variance. Removing it (w/o R_{rel}) leads to occasional factual deviations, as diverse paths lack a stabilizing anchor. Replacing the cluster-wise average with a signal derived from the single best-performing trajectory in each cluster (rel-max) yields inferior results. This maximum-accuracy formulation is brittle and prone to overfitting rare stochastic successes, whereas averaging effectively suppresses reward variance and yields a stable estimate of cluster reliability. By optimizing toward the expected cluster performance, the policy is encouraged to converge on reasoning patterns that demonstrate consistent factual correctness across multiple rollouts.

4.4 Analysis

Table 3: Correlation analysis between LLM-evaluator scores and reward-based rankings.

Metric	R_{acc} Only	$R_{\text{rel}} + R_{\text{acc}}$
Spearman ρ	0.39	**0.58**
Top-3 agreement (%)	21.7	**46.2**
Per-dimension correlations		
Logical Progression	0.32	**0.61**
Redundancy Avoidance	0.42	**0.53**
Factual Plausibility	0.35	**0.57**

Alignment with Expert-Ranked Query Quality. To validate the effectiveness of the reliability reward, we evaluate its alignment with high-quality query strategies using a strong LLM (Qwen3.6-27B). For each rollout, we extract the generated query sequences and rank them based on a multi-dimensional quality rubric covering clarity, logical progression, knowledge base alignment, redundancy avoidance, and factual plausibility (the complete evaluation prompt is provided in Figure 6). To ensure unbiased statistical comparison, ties in scores are resolved via random permutation. As shown in Table 3, incorporating R_{rel} substantially improves global alignment with the LLM evaluator, increasing Spearman's ρ from 0.39 to 0.58 and boosting Top-3 rank agreement by 24.5%. Furthermore, dimension-wise analysis reveals that R_{rel} most strongly correlates with logical progression and factual plausibility, confirming that it successfully steers the policy away from redundant or hallucinated queries toward coherent, step-wise valid reasoning paths. This demonstrates that reliability guidance provides multi-dimensional optimization signals that complement pure accuracy maximization.

Query Strategy Diversity and Policy Collapse Prevention. We investigate how R_{div} shapes the distribution of generated query strategies using semantic clustering. Specifically, we measure the semantic strategy entropy (see Appendix C for the definition and computation details), which quantifies the diversity of semantically distinct search patterns per question. As shown in Figure 3, the full model maintains a higher entropy distribution, with a substantial proportion of rollout groups sustaining moderate-to-high entropy ($H \geq 2$). In contrast, ablating R_{div} causes the distribution to concentrate into a few dominant strategy clusters, reflecting premature policy collapse.

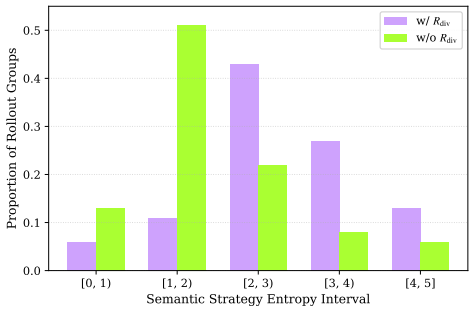


Figure 3: Distribution of rollout groups across semantic strategy entropy intervals. The model's strategy space contracts upon removal of the diversity reward, leading to reduced entropy concentration.

Fixed Retrieval Granularity vs. Query-Aware Adaptive Agentic Search. To evaluate the effectiveness of the adaptive continuation module, we compare it with fixed-granularity strategies ($k \in \{1, \ldots, 6\}$) across four benchmarks. As reported in Table 4, increasing k from 1 to 4 yields a steady gain in EM scores. This enhancement stems from the progressively richer contextual cues leveraged during each inference step. However, beyond $k = 4$, performance plateaus or degrades as context windows saturate with redundant passages, diluting the reasoning signal. Notably, $k = 1$ struggles with complex dependencies, while $k \geq 5$ suffers from information overload across all datasets. In contrast, the adaptive continuation module dynamically modulates document intake per step, consistently achieving the highest EM across all benchmarks.

Table 4: Comparison between fixed retrieval granularity (k) and Query-Aware Adaptive Agentic Search across four QA datasets.

Strategy	NQ	PopQA	HotpotQA	2Wiki
Fixed-k ($k = 1$)	51.2	43.7	46.3	46.2
Fixed-k ($k = 2$)	54.7	46.8	48.2	46.6
Fixed-k ($k = 3$)	56.1	47.3	49.8	48.6
Fixed-k ($k = 4$)	56.7	48.8	50.1	49.8
Fixed-k ($k = 5$)	56.3	47.9	49.3	49.5
Fixed-k ($k = 6$)	55.9	47.5	48.9	48.8
AdaDR-RAG	**58.3**	**50.1**	**52.5**	**51.6**

From Appendix D to Appendix I, we assess sensitivity to the trajectory-clustering threshold and tolerance-band width. We also examine query-conditioned retrieval control, training-seed stability, supporting-fact quality, retrieval efficiency, zero-shot domain transfer, trajectory grouping, and search-horizon sensitivity. Appendix K examines one successful case and one failure case.

5 Conclusion

We present **AdaDR-RAG**, an adaptive Agentic RAG framework that jointly optimizes retrieval depth and query strategies through diversity- and reliability-guided policy learning. By integrating implicit query comparison for dynamic context paging, trajectory clustering to capture and distinguish diverse search patterns, and a structured policy optimization framework, AdaDR-RAG provides comprehensive trajectory-level supervision that effectively mitigates policy collapse and stabilizes long-horizon multi-step reasoning. Extensive experiments across diverse benchmarks demonstrate that AdaDR-RAG consistently outperforms strong baselines, achieving superior answer accuracy and significantly enhanced interaction stability.

6 Limitations and Future Work

While AdaDR-RAG effectively balances information sufficiency and redundancy through adaptive continuation and enhances reasoning stability via trajectory clustering, two limitations warrant further investigation. First, the tolerance band boundaries are statically configured, which may not optimally align with the varying exploration-exploitation demands of queries across different complexity levels. Future work could explore dynamic threshold calibration mechanisms that adjust the tolerance band based on real-time query difficulty estimation. Second, our current clustering strategy relies primarily on sequential similarity of query trajectories, potentially overlooking inherent hierarchical or logical dependencies between reasoning steps. Incorporating structure-aware representations (e.g., tree-based or graph-aligned similarity metrics) into the clustering objective may further refine trajectory grouping and yield more granular optimization signals for policy learning.

AI USE STATEMENT

In this work, we used generative AI tools solely to refine the manuscript's language for clarity, fluency, and academic tone. We reviewed all suggested revisions to ensure that the technical meaning was preserved.

ETHICS STATEMENT

We confirm that this work adheres to established ethical research practices. All experiments were conducted using publicly available question-answering benchmarks, retrieval corpora, and pre-trained models, all of which are appropriately cited in the paper. No human participants were involved in this study.

REPRODUCIBILITY STATEMENT

Section 3 defines the adaptive retrieval controller, trajectory clustering, and reward functions, while Section 4 describes the experimental protocol and baselines. Appendix A documents the datasets, Appendix B reports implementation settings, and Appendix L provides training pseudocode. Additional validation appears in Appendix F to Appendix I.

REFERENCES

Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, Scott McCarley, Michael McCawley, Mohamed Nasr, Lin Pan, Cezar Pendus, John Pitrelli, Saurabh Pujar, Salim Roukos, Andrzej Sakrajda, Avi Sil, Rosario Uceda-Sosa, Todd Ward, and Rong Zhang. The TechQA dataset. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1269–1278, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.117. URL https://aclanthology.org/2020.acl-main.117/.

Yiqun Chen, Lingyong Yan, Weiwei Sun, Xinyu Ma, Yi Zhang, Shuaiqiang Wang, Dawei Yin, Yiming Yang, and Jiaxin Mao. Improving retrieval-augmented generation through multi-agent reinforcement learning. *arXiv preprint arXiv:2501.15228*, 2025.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2567–2577, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1259. URL https://aclanthology.org/D19-1259/.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Zixuan Ke, Weize Kong, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. Bridging the preference gap between retrievers and LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10438–10451, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.562. URL https://aclanthology.org/2024.acl-long.562/.

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. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

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. Retrieval-augmented generation for knowledge-intensive NLP tasks. In

Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5420–5438, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.276. URL https://aclanthology.org/2025.emnlp-main.276/.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.378. URL https://aclanthology.org/2023.findings-emnlp.378/.

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. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=dHng2O0Jjr.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.557. URL https://aclanthology.org/2023.acl-long.557/.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2024. URL https://arxiv.org/abs/2212.03533.

Peilin Wu, Mian Zhang, Kun Wan, Wentian Zhao, Kaiyu He, Xinya Du, and Zhiyu Chen. Hiprag: hierarchical process rewards for efficient agentic retrieval augmented generation. *arXiv preprint arXiv:2510.07794*, 2025.

Tianle Xia, Ming Xu, Lingxiang Hu, Yiding Sun, Wenwei Li, Linfang Shang, Liqun Liu, Peng Shu, Huan Yu, and Jie Jiang. Search-p1: Path-centric reward shaping for stable and efficient agentic rag training. *arXiv preprint arXiv:2602.22576*, 2026.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Tianhua Zhang, Kun Li, Junan Li, Yunxiang Li, Hongyin Luo, Xixin Wu, James Glass, and Helen Meng. Treeps-rag: Tree-based process supervision for reinforcement learning in agentic rag. *arXiv preprint arXiv:2601.06922*, 2026.

A DATASETS

We evaluate our framework on seven widely adopted question-answering benchmarks, encompassing both multi-hop reasoning and general knowledge retrieval tasks. The multi-hop datasets include HotpotQA, 2WikiMultihopQA, MuSiQue, and Bamboogle, while TriviaQA, PopQA, and Natural Questions (NQ) serve as benchmarks for single-hop factual QA. Following previous work (Zhang et al., 2026), we construct a unified training corpus of 12,000 examples by sampling 10,000 instances from HotpotQA and 2,000 from NQ. Evaluation is conducted on the complete held-out test sets across all seven datasets to comprehensively assess cross-domain generalization. Detailed statistics are summarized in Table 5.

Table 5: Dataset statistics. Training is performed on a unified subset of 12K examples from HotpotQA and NQ, while evaluation utilizes the full held-out test sets across all seven benchmarks.

Dataset	**Train Set Size**	**Test Set Size**
HotpotQA	10,000	7,405
2WikiMultihopQA	-	12,576
MuSiQue	-	2,417
Bamboogle	-	125
TriviaQA	-	11,313
PopQA	-	14,267
NQ	2,000	3,610
Total	**12,000**	**51,713**

B IMPLEMENTATION DETAILS

In the training phase, we employed the Group Relative Policy Optimization (GRPO) algorithm to fine-tune the `Qwen2.5-3B-Instruct` and `Qwen2.5-7B-Instruct`. For each training instance, trajectory sampling was conducted with model-specific configurations: 32 rollouts per prompt for the 3B-parameter variant and 16 rollouts per prompt for the 7B-parameter variant, reflecting a computational trade-off commensurate with model scale. Optimization was performed with a batch size of 512, and training proceeded for 500 GRPO update iterations. During sequence generation, we set the maximum input length and maximum output length to 2048 and 1024, respectively. The adaptive search module operates within a maximum reasoning horizon of 4 steps. The clustering similarity threshold δ is set to 0.7. Clustering uses only current-step rollouts for the first 50 GRPO updates, after which historical caching begins. The exploration-exploitation balancing reward is configured with a penalty scale of $\sigma = 0.1$ and a base weight of $\lambda_{bal} = 0.3$. Additionally, the minimum trajectory length threshold in the balancing reward is set to $L_{\min} = 2$. All experiments are conducted on 8 NVIDIA A100 (80GB) GPUs.

C SEMANTIC STRATEGY ENTROPY

For each question, we collect G query trajectories from the rollout group and partition them into I strategy clusters $\{\mathcal{C}_1, \ldots, \mathcal{C}_I\}$ using the clustering pipeline detailed in Section 3.4. Let n_i denote the number of trajectories assigned to cluster $\mathcal{C}_i$. The empirical probability of strategy i is $p_i = n_i/G$. The semantic strategy entropy is computed as:

$$H = -\sum_{i=1}^{I} p_i \log_2 p_i. \tag{10}$$

H measures the effective breadth of the agent's exploration. A higher H indicates that rollouts are distributed across multiple semantically distinct reasoning paths, reflecting sustained diversity. Conversely, a low H signifies that the policy has collapsed to a narrow set of template-like queries, regardless of surface-level string variations. In our analysis, entropy statistics are aggregated starting from training step 50. This warm-up period allows the policy to complete initial exploration and stabilize the diversity distribution, ensuring that the reported proportions reflect mature strategy dynamics rather than transient initialization noise.

D Clustering Sensitivity to Similarity Threshold Variations

We analyze the sensitivity of trajectory grouping to the similarity threshold δ in Table 6. A low threshold ($\delta \leq 0.5$) over-merges structurally distinct paths, inflating cluster sizes and diluting the intra-cluster accuracy signal R_{rel} with incorrect trajectories, which weakens policy guidance. Conversely, a high threshold ($\delta \geq 0.8$) fragments valid reasoning patterns into singletons or micro-clusters, reducing the statistical robustness of the accuracy estimation and over-amplifying the diversity reward R_{div}, leading to unstable exploration. The optimal setting ($\delta = 0.7$) balances structural cohesion and granularity, yielding moderate cluster sizes that maximize the joint optimization of R_{rel} and R_{div}. Notably, performance remains stable across $\delta \in [0.6, 0.8]$, demonstrating that the clustering mechanism is robust to moderate threshold variations and does not require fine-grained tuning.

Table 6: Effect of similarity threshold δ on model performance. Results remain stable across $\delta \in [0.6, 0.8]$, with $\delta = 0.7$ optimally balancing cluster cohesion and granularity.

δ	NQ	PopQA	HotpotQA	2Wiki
0.4	56.8	48.5	50.7	49.3
0.5	57.5	49.3	51.3	50.4
0.6	58.0	**50.6**	52.2	51.3
0.7	**58.3**	50.1	**52.5**	**51.6**
0.8	57.7	49.7	51.8	51.2
0.9	57.2	48.9	51.3	50.6

E Impact of Tolerance Bandwidth on Exploration-Exploitation Balance

We center the tolerance band at $r = 0.5$, as it represents the theoretical equilibrium between query deepening (exploitation of cached results) and semantic switching (exploration of new sub-goals). By symmetrically expanding the bandwidth from $[0.5, 0.5]$ to $[0.1, 0.9]$, we isolate the effect of constraint tightness on policy stability without introducing directional bias. We systematically evaluate the sensitivity of R_{bal} to the tolerance band $[r_{\text{low}}, r_{\text{high}}]$ across five configurations: $[0.5, 0.5]$, $[0.4, 0.6]$, $[0.3, 0.7]$, $[0.2, 0.8]$, and $[0.1, 0.9]$.

The results in Figure 4 reveal a clear non-monotonic relationship between bandwidth and final EM score. The point-target setting ($[0.5, 0.5]$) yields the lowest performance, as the zero-width band forces the policy to rigidly maintain exactly 50% repetition, severely restricting adaptive depth control and penalizing legitimate query pivots. Narrowing to $[0.4, 0.6]$ improves training stability but still constrains natural reasoning flows. The $[0.3, 0.7]$ configuration achieves peak performance, providing sufficient flexibility for the agent to dynamically deepen or switch per query while effectively suppressing degenerate loops ($r > 0.7$) and excessive thrashing ($r < 0.3$). Expanding further to $[0.2, 0.8]$ and $[0.1, 0.9]$ gradually degrades accuracy, as the widened safe zone weakens the exponential penalty on pathological behaviors, allowing the policy to drift toward redundant paging or shallow exploration. This demonstrates that the tolerance band acts as a *soft regularization mechanism*: an optimally calibrated width (≈ 0.4) preserves policy adaptivity while maintaining a clear boundary against reward-hacking behaviors, whereas extreme widths either over-constrain exploration or fail to penalize structural collapse.

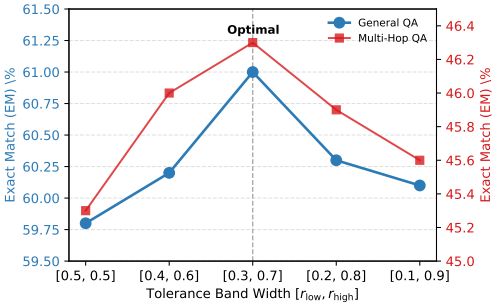


Figure 4: Evaluating performance under varying tolerance bandwidth $[r_{\text{low}}, r_{\text{high}}]$. A moderate bandwidth ($[0.3, 0.7]$) achieves optimal EM by balancing query deepening and semantic switching, while overly narrow or wide bands restrict policy adaptivity or fail to penalize degenerate search behaviors.

F Validation of Query-Conditioned Retrieval Control

F.1 Observed Query Repetition across Different Methods

For trajectories with $L > 1$ search actions, we define the consecutive repetition ratio as $r = n_{\text{rep}}/(L-1)$, where n_{rep} counts transitions between identical consecutive queries. Table 7 reports the mean r for each method and the percentage of trajectories with $r > 0.7$. Search-P1 and TreePS-RAG exhibit lower exact-query repetition, although repetition does not serve as an explicit retrieval-depth action in these methods. Within AdaDR-RAG, R_{bal} reduces the mean repetition ratio from 0.37 to 0.28 and the proportion of high-repetition trajectories from 8.2% to 6.3%. These results indicate that it suppresses excessive repetition without eliminating legitimate retrieval deepening.

Table 7: Consecutive-query repetition across agentic retrieval methods.

Method	Mean r	$r > 0.7$ (%)
Search-P1	0.12	4.3
TreePS-RAG	0.08	3.7
AdaDR-RAG	0.28	6.3
w/o R_{bal}	0.37	8.2

F.2 Effects of Query-Matching Strategies

To examine how the criterion for identifying repeated queries affects cache reuse and answer accuracy, we compare raw exact matching with canonicalized and semantic matching. Canonicalized matching lowercases and normalizes whitespace and punctuation. Semantic matching reuses the cache when consecutive-query embedding similarity is at least τ, using the same `all-MiniLM-L6-v2` encoder as trajectory clustering.

Table 8: Effect of query-matching strategies on question-answering accuracy with Qwen2.5-7B-Instruct. Results are reported as exact match (EM).

Controller	NQ	PopQA	HotpotQA	2Wiki
Raw exact matching (Ours)	58.3	50.1	52.5	51.6
Canonicalized exact matching	**58.7**	50.1	52.8	**51.8**
Semantic matching ($\tau = 0.95$)	58.0	**50.4**	**53.0**	51.2
Semantic matching ($\tau = 0.90$)	57.6	49.8	51.9	51.2

As shown in Table 8, canonicalized exact matching improves average EM by only 0.2 points, suggesting that normalization handles minor lexical variations without materially affecting performance. Semantic matching with $\tau = 0.95$ remains comparable to raw exact matching, while lowering the threshold to $\tau = 0.90$ causes a 0.5-point average drop. This suggests that a more permissive threshold can incorrectly merge semantically related but functionally distinct sub-goals, leading the controller to continue paging through an inappropriate cache. Given its added embedding cost and limited gains, we retain raw exact matching as the default.

G Statistical Stability and Evidence-Level Evaluation

G.1 Three independent training seeds

We repeat the training of AdaDR-RAG and the strongest RL-based baselines using three independent random seeds. Evaluation uses the same test sets, retrieval index, and decoding configuration for every seed. Table 9 reports the mean and standard deviation of EM scores. AdaDR-RAG achieves the highest mean EM on each dataset with consistently low standard deviations of 0.2–0.3 points. The gains over the strongest baselines exceed the observed variation across random seeds, supporting the stability of the improvements.

Table 9: Question-answering performance across three independent training runs. Results report the mean and standard deviation of EM scores.

Method	NQ	PopQA	HotpotQA	2Wiki
Search-P1	56.6 ± 0.2	47.5 ± 0.2	43.3 ± 0.4	39.6 ± 0.4
TreePS-RAG	52.4 ± 0.3	48.3 ± 0.4	50.7 ± 0.2	48.5 ± 0.4
AdaDR-RAG	$\mathbf{58.5 \pm 0.2}$	$\mathbf{50.1 \pm 0.3}$	$\mathbf{52.5 \pm 0.3}$	$\mathbf{51.4 \pm 0.2}$

G.2 Answer evaluation using exact match and token-level F1

We supplement EM with standard token-level F1, which captures partial answer overlap. As shown in Table 10, AdaDR-RAG achieves the highest average scores under both metrics across the seven benchmarks. Relative to TreePS-RAG, it improves average EM and F1 by 3.6 and 4.4 points, respectively. The F1 gain indicates that the improvements extend beyond exact reference-string matching.

Table 10: Average EM and F1 across seven QA benchmarks.

Method	Avg. EM	Avg. F1
Search-P1	47.3	53.6
TreePS-RAG	49.0	54.9
AdaDR-RAG	**52.6**	**59.3**

G.3 Supporting-fact retrieval and answer grounding

To evaluate retrieval quality independently from answer correctness, we use HotpotQA supporting-fact annotations. We extract the documents retrieved along the search trajectories and evaluate the recovered supporting facts against the annotations using supporting-fact EM/F1. We also report joint EM/F1 to measure evidence grounding and answer correctness jointly. As shown in Table 11, AdaDR-RAG improves supporting-fact EM/F1 over TreePS-RAG by 3.9/4.2 points and achieves higher joint EM/F1. These results indicate better evidence retrieval and stronger alignment between supporting evidence and final answers.

Table 11: Answer and supporting-fact scores on HotpotQA.

Metric	TreePS-RAG	AdaDR-RAG
Answer EM	50.7	**52.5**
Answer F1	56.9	**59.4**
Support EM	27.3	**31.2**
Support F1	64.7	**68.9**
Joint EM	22.6	**24.7**
Joint F1	45.0	**48.5**

H Retrieval Efficiency and Transfer

H.1 Search actions, engine calls, and injected context

We evaluate accuracy and retrieval efficiency on NQ, PopQA, HotpotQA, and 2Wiki under the same Qwen2.5-7B-Instruct backbone, E5 retriever, 2018 Wikipedia corpus, and inference environment. Search actions count agent-generated search operations and also reflect trajectory length. Unique engine calls count queries actually submitted to the retriever, excluding consecutive repeated-query actions served from cache. Injected chunks count retrieved passages added to the context. Latency is measured end to end.

Following their original retrieval interfaces, Search-P1 and TreePS-RAG inject three chunks per action, whereas AdaDR-RAG injects one cached chunk per action for finer-grained evidence control. Table 12 reports the results averaged over the four benchmarks.

AdaDR-RAG achieves the highest average EM while injecting substantially fewer chunks and obtaining the lowest end-to-end latency. Although it generates more search actions, consecutive repeated-query actions reuse cached rankings rather than invoking the search engine again. This enables finer-grained evidence acquisition while reducing external retrieval overhead and context redundancy. The gains are therefore unlikely to stem from increased retrieval volume.

Table 12: Question-answering accuracy and retrieval efficiency averaged over NQ, PopQA, HotpotQA, and 2Wiki.

Method	Avg. EM	Search actions	Engine calls	Injected chunks	Latency (s)
Search-P1	46.7	**2.6**	**2.6**	7.9	8.1
TreePS-RAG	49.9	3.2	3.2	9.6	10.7
AdaDR-RAG	**53.1**	3.7	2.9	**3.7**	**6.9**

H.2 Generalization across retrievers

To evaluate whether AdaDR-RAG is tied to a specific retriever, we additionally train and test with BM25, Contriever, and `gte-modernbert-base` under the same backbone, training data, corpus, and evaluation protocol. As shown in Table 13, performance remains strong across retrievers, while `gte-modernbert-base` provides consistent gains over E5. These results indicate that AdaDR-RAG is compatible with both sparse and dense retrievers.

Table 13: Question-answering accuracy with sparse and dense retrievers under a shared training and evaluation protocol. Results are reported as EM.

Retriever	NQ	PopQA	HotpotQA	2Wiki
BM25	55.6	46.3	51.1	48.4
Contriever	56.2	48.7	52.3	50.7
E5	58.3	50.1	52.5	51.6
gte-modernbert-base	**59.6**	**50.6**	**54.4**	**53.2**

H.3 Additional domain-transfer evaluation

We evaluate zero-shot transfer on TechQA (Castelli et al., 2020), which contains technical-support questions and candidate technical documents, and PubMedQA (Jin et al., 2019), which contains biomedical questions grounded in PubMed abstracts. The policy trained on NQ and HotpotQA is evaluated without domain-specific training using the same Qwen2.5-7B configuration. We report token-level F1 for TechQA and accuracy for PubMedQA. As shown in Table 14, AdaDR-RAG consistently outperforms both baselines, indicating transfer beyond open-domain QA to technical-support and biomedical questions.

Table 14: Zero-shot domain transfer to TechQA (F1) and PubMedQA (accuracy).

Method	TechQA	PubMedQA
Search-P1	56.2	84.3
TreePS-RAG	57.9	86.7
AdaDR-RAG	**61.2**	**89.5**

I Trajectory Structure and Search-Horizon Sensitivity

I.1 Length-sensitive and length-tolerant grouping

In the default clustering rule (Equation 3), the equal-length indicator limits comparison to synchronized search steps, while dense query embeddings allow paraphrases at corresponding positions. Trajectories with different numbers of retrieval decisions may differ in retrieval depth, evidence order, or sub-goal decomposition. Comparing them without alignment can therefore merge distinct strategies. A length difference can also arise when one trajectory adds a refinement step without changing its underlying strategy. We therefore evaluate a length-tolerant variant based on monotonic semantic alignment with a gap penalty:

$$S_{\text{align}}(X^{(i)}, X^{(j)}) = \max_{\mathcal{A}} \frac{1}{|\mathcal{A}|} \sum_{(u,v)\in\mathcal{A}} \cos(e_u^{(i)}, e_v^{(j)}) - \lambda_{\text{gap}} \frac{|L_i - L_j|}{\max(L_i, L_j)}, \tag{11}$$

where $\mathcal{A}$ is a nonempty monotonic alignment between query steps and $\lambda_{\text{gap}} = 0.2$. This formulation allows trajectories with different lengths to be compared while penalizing unmatched steps. As shown in Table 15, monotonic alignment slightly improves HotpotQA but reduces average performance across the other datasets. This suggests that length differences can reflect meaningful structural variations, such as retrieval depth. Cross-length alignment may therefore reduce cluster purity and weaken cluster-based rewards, supporting the default equal-length, step-wise formulation.

Table 15: Effect of trajectory alignment on question-answering accuracy.

Trajectory similarity	NQ	PopQA	HotpotQA	2Wiki
Equal length + step-wise (Ours)	**58.3**	**50.1**	52.5	**51.6**
Monotonic alignment + gap penalty	57.6	49.7	**52.9**	50.8

I.2 Maximum reasoning horizon

The sensitivity to the horizon is investigated under the same training and inference settings.

As shown in Table 16, increasing T from 2 to 4 improves performance on all four datasets. Longer horizons provide consistent but modest additional gains, with larger improvements on multi-hop datasets. These results suggest that additional search capacity is useful for deeper evidence accumulation. Here, T specifies the maximum horizon, while individual trajectories may terminate earlier.

Table 16: Effect of the maximum search horizon.

Horizon	NQ	PopQA	HotpotQA	2Wiki
$T = 2$	55.2	46.3	47.4	46.9
$T = 4$	58.3	50.1	52.5	51.6
$T = 6$	58.8	**50.4**	53.2	52.2
$T = 8$	**59.1**	**50.4**	**53.4**	**52.5**

I.3 Sensitivity of controller and clustering hyperparameters

We evaluate the method-specific hyperparameters while holding all other settings fixed. Table 17 reports average EM over NQ, PopQA, HotpotQA, and 2Wiki. The maximum variation is 0.7 EM points, indicating limited sensitivity within the evaluated ranges. The tolerance band remains manually specified, motivating automatic calibration in future work.

Table 17: Sensitivity to controller and clustering hyperparameters. EM scores are averaged over NQ, PopQA, HotpotQA, and 2Wiki.

Parameter	Values	Avg. EM	Maximum variation
Clustering threshold δ	0.6 / 0.7 / 0.8	53.0 / **53.1** / 52.6	0.5
Tolerance band	$[0.2, 0.8]$ / $[0.3, 0.7]$ / $[0.4, 0.6]$	52.6 / **53.1** / 52.4	0.7
λ_{bal}	0.1 / 0.3 / 0.5	52.8 / **53.1** / 52.6	0.5
σ	0.05 / 0.10 / 0.20	52.5 / **53.1** / 52.8	0.6

J Prompt for Agentic RAG

Figure 5 details the system prompt that orchestrates the agent's interactive reasoning loop. It enforces a structured output format using `<search>`, `<information>`, and `<answer>` tags to guide the model through iterative deliberation, query formulation, and evidence synthesis. Crucially, the prompt explicitly authorizes the agent to either repeat identical queries to paginate through additional results or reformulate/decompose queries to adjust search scope, directly enabling the adaptive retrieval and dynamic depth control mechanisms formalized in Section 3.3.

Agentic RAG Prompt

Answer the given question. You must conduct reasoning inside <think> and </think> first every time you get new information. After reasoning, if you find you lack some knowledge, you can call a search engine by <search> query </search> and it will return the top searched results between <information> and </information>. If the search results are insufficient to answer this query, you can call the search engine using the same query. You can also rephrase or break down the query to make the search more accurate. You can search as many times as you want. If you find no further external knowledge needed, you can directly provide the answer inside <answer> and </answer>, without detailed illustrations. For example, <answer> Beijing </answer>.

Figure 5: The prompt used in agentic rag.

Evaluation Prompt for Multi-Dimensional Query Trajectory Quality

You are an expert evaluator of multi-hop reasoning and agentic retrieval systems. Your task is to score the quality of a step-wise query sequence generated during a reasoning rollout.

Please evaluate the sequence across the following dimensions (1–5 scale, where 5 is excellent):
1. Clarity & Specificity: Are queries unambiguous, precise, and free from vague or open-ended phrasing?
2. Logical Progression: Does each query naturally follow from the previous step and advance the reasoning toward the final answer?
3. Knowledge Base Alignment: Are queries phrased in a standard, search-friendly format that aligns with common external knowledge sources (e.g., canonical entity names, Wikipedia titles)?
4. Redundancy Avoidance: Does the sequence avoid repetitive, circular, or unnecessary queries that yield no new information?
5. Factual Plausibility: Do queries avoid hallucinated constraints, impossible assumptions, or logically contradictory premises?
Input:
- Original Question: {question}
- Query Sequence: {query_sequence}
Output your evaluation in strict JSON format:
{
“dimension_scores”: {“clarity”: x, “progression”: x, “alignment”: x, “redundancy”: x, “plausibility”: x},
“overall_score”: x,
“rationale”: “1-2 sentences summarizing key strengths and weaknesses.”
}
Do not output any additional text or markdown.

Figure 6: The structured prompt instructs the LLM evaluator to score generated query sequences across five criteria: clarity & specificity, logical progression, knowledge base alignment, redundancy avoidance, and factual plausibility.

K Case Study

We provide two cases in Figures 7 and 8.

Case 1. To illustrate the practical impact of our design, we examine a multi-hop query requiring precise temporal comparison: *“Which film has the director who was born first, Yaadein (1964 Film) or The Student’s Romance?”* (Ground Truth: *The Student’s Romance*). As shown in the interaction traces, Search-R1 generates generic queries (e.g., “director of The Student’s Romance”) that omit

the critical "birth year" constraint. Consequently, it incorrectly associates the target with an irrelevant entity (*Student Seduction*), fails to extract temporal information, and halts exploration after retrieving the birth year of Sunil Dutt (1929) for *Yaadein*. This highlights the brittleness of fixed retrieval schedules when queries lack essential constraints. In contrast, AdaDR-RAG explicitly incorporates "the birth year" constraint into its search terms. When initial evidence proves insufficient, the model autonomously repeats the targeted query to fetch subsequent cached batches, successfully identifying the correct director, Arthur Barron (born in 1928). It then explicitly compares this birth year with that of Sunil Dutt (1929) before generating the final answer.

Case 2. We examine a multi-constraint query: *"Fireworks that burn red contain which element, the only one named after a place in UK?"* (Ground Truth: *Strontium*). As shown in the interaction traces, AdaDR-RAG initiates an over-constrained search query that packs multiple attributes into a single string. This leads to poor retrieval precision, as the search engine returns commercial entities (e.g., firework companies) rather than chemical data. When initial evidence proves insufficient, the model attempts minor query rephrasing but fails to systematically decompose the constraints. Instead of isolating the "geographical naming" or "flame color" attributes separately, it abruptly pivots to an unjustified query (*"sodium element brit naming"*). Critically, even after retrieving a document explicitly stating that sodium produces *yellow* flames, the model ignores this contradictory evidence and hallucinates the final answer. In contrast, Search-R1 successfully decomposes the problem by querying the geographical naming constraint, retrieving a candidate list that includes *Strontium*, and cross-validating the flame color property. This case highlights a practical limitation: when initial query generation over-packs constraints, it can degrade retrieval quality and disrupt the constraint verification pipeline, leading to factual contradictions. It underscores the necessity of explicit constraint decomposition or adaptive query prioritization for handling highly specific multi-attribute questions.

Case 1 (Successful Case)

Question Source: 2WikiMultihopQA
Question: Which film has the director who was born first, Yaadein (1964 Film) or The Student'S Romance?
Ground Truth: The Student's Romance

Generation Model: Qwen2.5-7B (Search-R1)
Response: <think> To determine which film has the director who was born first between "Yaadein" (1964 Film) and "The Student's Romance," I first need to identify the directors of these films. Yaadein is from 1964. Without direct access to historical director information, I'll start with "The Student's Romance." </think>
<search> director of The Student's Romance </search>
<information> Doc 1(Title: "Student Seduction") Student Seduction Student Seduction is a 2003 television film directed by Peter Svatek and starring Elizabeth Berkley... </information>
<think> The Student's Romance" is not a 1964 film. It refers to "Student Seduction" (2003), which clearly mentions the director as Peter Svatek. Without immediate information on "Yaadein," it suggests "Yaadein" could be from 1964. For Yaadein: </think>
<search> director of Yaadein (1964 film) </search>
<information> Doc 1(Title: "Yaadein (1964 film)") Yaadein (1964 film) Yaadein (English: ""Memories"") is a 1964 black and white Hindi film directed and produced by Sunil Dutt also starring himself... </information>
<think> The director of Yaadein (1964) is Sunil Dutt, born in 1929. The director of "The Student's Romance" (2003) is Peter Svatek, implying "Yaadein" was directed by a person born earlier. Therefore: Yaadein (1964 Film) had the director who was born first.

</think>
<answer> Yaadein (1964 Film) </answer>

Generation Model: Qwen2.5-7B (AdaDR-RAG)
Response: <think> Yaadein is a 1964 film, so I need to find out which one, Yaadein or The Student's Romance, has the director who was born first. To do this, I need to find out the birth year of the director of Yaadein and the birth year of the director of The Student's Romance. </think>
<search> director of Yaadein 1964 film birth year </search>
<information> Yaadein (1964 film) Yaadein (English: ""Memories"") is a 1964 black and white Hindi film directed and ...
</information>
<think> From the information, the director of Yaadein is Sunil Dutt, and he was born in 1929. </think>
<search> director of The Student's Romance birth year </search>
<information> Student Seduction Student Seduction is a 2003 television film directed by Peter Svatek and starring Elizabeth Berkley... </information>
<think> The director of Yaadein (1964) is Sunil Dutt, born in 1929. The director of "The Student's Romance" (2003) is Peter Svatek, implying "Yaadein" was directed by a person born earlier. Therefore: Yaadein (1964 Film) had the director who was born first.
</think>
<search> director of The Student's Romance birth year </search>
<information> It was the first film directed by Arthur Barron, and won the prize for Best First Work in the 1973 Cannes Film Festival... </information>
<think> The director of The Student's Romance is Arthur Barron, born in 1928. Sunil Dutt was born in 1929, while Arthur Barron was born in 1928. So Arthur Barron was born first. </think>
<answer> The Student's Romance </answer>

Figure 7: Case study comparison on a multi-hop query from 2WikiMultihopQA.

Case 2 (Failed Case)

Question Source: TriviaQA
Question: Fireworks that burn red contain which element, the only one named after a place in UK?
Ground Truth: Strontium

Generation Model: Qwen2.5-7B (Search-R1)
Response: <think> I need to know which element when fireworks burn red is named after a place in the UK. I do not have this specific knowledge available. </think>
<search> element named after a place in UK </search>
<information> Doc 1(Title: "Trivial name") gods for mankind. Discoverers of some elements named them after their home country or city...strontium, scandium, europium, thulium (after an old Roman name for the far north of Scandinavia), holmium, copper...
</information>
<think> Based on the information provided from the search, one

element of interest named after a place in the UK is polonium, named after Poland by Marie Curie, but it's not the correct one. I find strontium to be named after Strontian, Scotland, which matches the UK place criterion. Polonium is the element used in fireworks that burns red and is named after a place in the UK, Strontian, Scotland. However, Strontium is the actual element named after Strontian in Scotland. Therefore, the key here is to recognize that this specific problem refers to which element is red in fireworks which seems to be Strontium when compounds of the metal Sr are burned. But the original query seeks an element based on a UK place, and Strontium fits the place but not the initial key red coloring context directly. Conclusion based on the specific query then aligns with the direct answer provided by the typical reference: </think>

<answer> Strontium </answer>

Generation Model: Qwen2.5-7B (AdaDR-RAG)
Response: <think> I need to find out which element burns red in fireworks and if it is named after a place in the UK. </think>

<search> element burns red in fireworks named after a place in UK </search>

<information> Fantastic Fireworks Fantastic Fireworks is a British company that stages large-scale fireworks displays in Europe and the Far East. It was founded in 1985 and is based in Pepperstock. Fantastic Fireworks are twice winners of the British Fireworks Championship (1997 and 2015) and have been recognized for the quality of their fireworks displays... </information>

<think> Based on the information given, the search did not lead directly to the specific element. Let me restrict search: </think>

<search> element burns red in fireworks and is named after a British place </search>

<information> with burning sparklers on paper, which they called pyrografie. Since then the two artists have drawn more than 20,000 pyroportraits. Sparkler A sparkler is a type of hand-held firework that burns slowly while emitting colored flames, sparks, and other effects... </information>

<think> Go grab naming specifics: </think>

<search> sodium element brit naming </search>

<information> Latin name of ""sodanum"" was used as a headache remedy...Sodium imparts an intense yellow color to flames. As early as 1860, Kirchhoff and Bunsen noted the high sensitivity of a sodium flame test... </information>

<answer> sodium </answer>

Figure 8: Case study on a multi-constraint query illustrating a failure mode of AdaDR-RAG.

L TRAINING ALGORITHM

Algorithm 1 AdaDR-RAG: Single Training Iteration with GRPO

Require: Query batch Q, policies $\pi_\theta, \pi_{\theta_{old}}, \pi_{ref}$, group size G, horizon T, cap λ_{bal}, tolerance bounds r_{low}, r_{high}, scale σ, similarity threshold δ, learning rate η
Ensure: Updated policy π_θ

Initialize gradient accumulator $\nabla\mathcal{L} \leftarrow \mathbf{0}$
for each $q \in \mathcal{Q}$ **do**
  $y_q^* \leftarrow \text{GroundTruth}(q)$, $\mathcal{K} \leftarrow \emptyset$, $\mathcal{H}_q \leftarrow \text{LoadHistory}(q)$, $\mathcal{G} \leftarrow \emptyset$
  for $g = 1$ to G **do**
    $s_1 \leftarrow \text{InitState}(q)$, $X^{(g)} \leftarrow [\,]$, $x_0 \leftarrow \varnothing$, $k \leftarrow 1$
    for $t = 1$ to T **do**
      $a_t, x_t \sim \pi_\theta(\cdot \mid s_t)$, $X^{(g)} \leftarrow X^{(g)} \cup \{x_t\}$
      if $\text{IsFinalAnswer}(a_t)$ **or** $k \geq K_{\max}$ **then**
        break
      end if
      if $x_t = x_{t-1}$ **and** $x_{t-1} \neq \varnothing$ **then**
        $k \leftarrow k + 1$ ▷ Iterative deepening on cache
      else
        $k \leftarrow 1$
        if $x_t \notin \mathcal{K}$ **then**
          $\mathcal{K}[x_t] \leftarrow \text{SearchEngine}(x_t, \text{top-}M)$
        end if
      end if
      $\mathbf{b}_k \leftarrow \mathcal{K}[x_t][k]$, $\mathcal{B}_t \leftarrow \mathbf{b}_k$ ▷ Map to step-t context
      $s_{t+1} \leftarrow \text{Concat}(s_t, a_t, x_t, \mathcal{B}_t)$
    end for
    $y^{(g)} \leftarrow \text{ExtractAnswer}(a_{\text{final}})$, $\mathcal{G} \leftarrow \mathcal{G} \cup \{(X^{(g)}, y^{(g)})\}$
    $\mathcal{H}_q \leftarrow \mathcal{H}_q \cup \{X^{(g)}\}$ ▷ Update per-query trajectory buffer
  end for
  Compute the similarity matrix $\mathbf{S} = (S_{ij})$ over $\mathcal{H}_q$ using Eq. 3
  $\mathcal{C}_\ell \leftarrow \text{AgglomerativeCluster}(\mathcal{H}_q, \mathbf{S}, \text{threshold} = \delta)$
  $\mathcal{R} \leftarrow [\,]$
  for each $(X^{(g)}, y^{(g)}) \in \mathcal{G}$ **do**
    $\mathcal{C} \leftarrow \text{FindCluster}(\mathcal{C}_\ell, X^{(g)})$
    $n_{\text{rep}} \leftarrow \sum_{t=2}^{L_g} \mathbb{I}(x_t = x_{t-1})$, $r \leftarrow n_{\text{rep}} / \max(L_g - 1, 1)$
    $D(r) \leftarrow \max(0, r - r_{\text{high}}) + \max(0, r_{\text{low}} - r)$ ▷ Deviation outside tolerance band
    $R_{\text{bal}}^{(g)} \leftarrow \begin{cases} \lambda_{\text{bal}}, & L_g \leq L_{min} \\ \lambda_{\text{bal}} \cdot \exp\left(-\frac{D(r)^2}{2\sigma^2}\right), & L_g > L_{min} \end{cases}$ ▷ Balancing reward (Sec. 3.3)
    $R_{\text{div}}^{(g)} \leftarrow 1/\log(1 + |\mathcal{C}|)$, $R_{\text{rel}}^{(g)} \leftarrow \frac{1}{|\mathcal{C}|} \sum_{(X^{(j)}, y^{(j)}) \in \mathcal{C}} \mathbb{I}(y^{(j)} = y_q^*)$, $R_{\text{acc}}^{(g)} \leftarrow \mathbb{I}(y^{(g)} = y_q^*)$
    $R^{(g)} \leftarrow R_{\text{div}}^{(g)} + R_{\text{rel}}^{(g)} + R_{\text{acc}}^{(g)} + R_{\text{bal}}^{(g)}$
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{R^{(g)}\}$
  end for
  $\mu, \sigma \leftarrow \text{Mean}(\mathcal{R}), \text{Std}(\mathcal{R})$
  for each $g = 1$ to $|\mathcal{G}|$ **do**
    $A_g \leftarrow (R^{(g)} - \mu)/(\sigma + \gamma)$
    $\nabla\mathcal{L} \leftarrow \nabla\mathcal{L} + \nabla_\theta \mathcal{L}(X^{(g)}, A_g; \pi_{\theta_{\text{old}}}, \pi_{\text{ref}}, \beta_{\text{KL}}, \epsilon)$
  end for
end for
$\theta \leftarrow \theta + \eta \frac{1}{|\mathcal{Q}|} \nabla\mathcal{L}$, $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
return π_θ