GlobalRAG: Enhancing Global Reasoning in Multi-hop Question Answering via Reinforcement Learning

Anonymous ACL submission

Abstract

Reinforcement learning has recently shown promise in improving retrieval-augmented generation (RAG). Despite these advances, its effectiveness in multi-hop question answering (QA) remains limited by two fundamental limitations: (i) global planning absence to structure multi-step reasoning, and (ii) unfaithful execution, which hinders effective query formulation and consistent use of retrieved evidence. We propose **GlobalRAG**, a reinforcement learning framework designed to enhance global reasoning in multi-hop QA. GlobalRAG decomposes questions into subgoals, coordinates retrieval with reasoning, and refines evidence iteratively. To guide this process, we introduce *Planning Quality Reward* and *SubGoal Completion Reward*, which encourage coherent planning and reliable subgoal execution. In addition, a progressive weight annealing strategy balances process-oriented and outcome-based objectives. Extensive experiments on both in-domain and out-of-domain benchmarks demonstrate that GlobalRAG significantly outperforms strong baselines while using only 8k training data (42% of the training data used by strong baselines), achieving average improvements of 14.2% in both EM and F1.

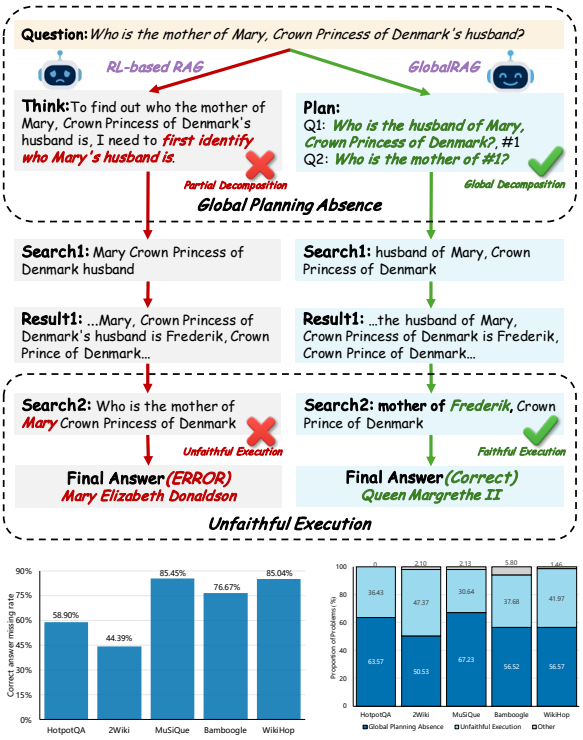

Figure 1: (*top*) Example of global planning absence and unfaithful execution. (*bottom left*) The most frequent failure of Search-R1 is missing the correct document. (*bottom right*) Error distribution where these two issues account for over 94% of failures.

1 Introduction

Recently, reinforcement learning with verifiable rewards (RLVR) has achieved rapid progress (Guo et al., 2025; Dong et al., 2025; Lai et al., 2025), particularly in retrieval-augmented generation (RAG), where it has delivered highly competitive results (Jin et al., 2025a; Chen et al., 2025; Song et al., 2025a). However, while RL-based methods have shown strong performance on single-hop questions, significant challenges remain in handling multi-hop and complex questions. Recent approaches such as *StepSearch* (Wang et al., 2025), *LeTS* (Zhang et al., 2025), *ParallelSearch* (Zhao et al., 2025), and *GraphRAG-R1* (Yu et al., 2025) attempt to encourage models to invoke the retriever more frequently, thereby alleviating *shallow retrieval problems*.

Despite these advances, the underlying causes of failure in multi-hop QA remain unresolved. In particular, multi-hop QA continues to suffer from two fundamental limitations: **global planning absence**, where the model fails to construct an overall plan for decomposing and solving the question (Lu et al., 2025; Mei et al., 2025), and **unfaithful execution**, where retrieval drifts from the original target and results in missing the correct document (Zhu et al., 2025). Figure 1(top) illustrates a real case: the RL-based model fails to construct a global plan, only partially decomposes the problem, and eventually

loses the original target—an instance of unfaithful execution—failing to identify the correct subgoal answer *"Frederik"*.

Building on these observations, we conducted a quantitative study. Specifically, we sampled 300 examples from each of five authoritative multi-hop reasoning benchmarks: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), Bamboogle (Press et al., 2023) and WikiHop (Welbl et al., 2018). We first examined the major causes of failures in multi-hop QA. As shown in Figure 1(bottom left), the most frequent failure arises from the inability of the model to retrieve documents containing the correct answer after multiple retrieval attempts, accounting for 44.39%–85.45% of errors across datasets. We then conducted a finer-grained analysis of the *Correct Answer Missing* problem by inspecting reasoning trajectories with *Deepseek-R1* (Guo et al., 2025). As illustrated in Figure 1(bottom right), the most severe error types are the **global planning absence** and the **unfaithful execution**, which together account for more than 94% of failures (further details are provided in Appendix B). These two error types constitute the primary sources of the *Correct Answer Missing* issue, which in turn further disrupts the overall reasoning chain in multi-hop QA.

To this end, we propose **GlobalRAG**, a reinforcement learning framework designed to enhance global reasoning in multi-hop QA. To address the issue of *global planning absence*, GlobalRAG introduces planning-aware optimization by explicitly decomposing questions into subgoals, inspired by prior work PSE (Xu et al., 2025) that leverages graph structures to enhance reasoning. To guide this process, we design a *Planning Quality Reward*, which evaluates the generated plan based on both graph-structure similarity and semantic similarity. Combined with GRPO (Shao et al., 2024) optimization, this enables the model to maintain a more coherent and global perspective when reasoning over complex questions. To further mitigate the *unfaithful execution*, GlobalRAG incorporates a *SubGoal Completion Reward* that encourages the model to faithfully follow the global plan. This reward integrates reasoning and retrieval, guiding the model to dynamically complete each subgoal in sequence and ensuring consistent alignment between planned subgoals and executed reasoning steps. Finally, inspired by TIRESRAG-R1 (He et al., 2025), we adopt a *progressive weight annealing* strategy to balance process-oriented and outcome-based objectives during training, thereby stabilizing early structural learning while ensuring convergence to high-quality final answers.

Experiments across multiple benchmark datasets show that our graph-guided dense supervision significantly improves multi-hop reasoning performance, validating the effectiveness of our approach.

Our contributions can summrize as follows:

- We identify two fundamental limitations of existing RL-based approaches for multi-hop QA: global planning absence and unfaithful execution.
- We propose GlobalRAG, a reinforcement learning framework that introduces planning-aware optimization to strengthen global reasoning in multi-hop QA.
- Extensive experiments on both in-domain and out-of-domain datasets demonstrate that GlobalRAG consistently outperforms strong baselines, yielding average gains of 14.2% in both EM and F1.

2 Related Work

Retrieval-Augmented Generation for Multi-hop QA. Multi-hop QA addresses questions that require multi-step reasoning over evidence from multiple sources to produce a complete answer (Li and Du, 2023; Zhang et al., 2024; Cheng et al., 2025). Retrieval-augmented generation (RAG) addresses multi-hop QA by coupling retrieval with generation (Ji et al., 2023; Feng et al., 2024; Shi et al., 2024). Existing RAG for multi-hop QA primarily follows two lines: graph-structured approaches (Gutiérrez et al., 2025; Gutiérrez et al., 2025; Fang et al., 2025), which map textual knowledge to an entity–relation graph and aggregate evidence along multi-hop paths to answer multi-hop questions; and iterative RAG approaches (Trivedi et al., 2023; Shao et al., 2023; Liu et al., 2024; Jiang et al., 2025; Li et al., 2025), which address multi-hop QA by performing multiple steps of retrieval and reasoning (e.g., IRCOT (Trivedi et al., 2023) is interleaved step by step to reduce information omission, Search-o1 (Li et al., 2025)) embeds agent-style search processes into large reasoning models). However, graph-structured approaches require building and maintaining the graph (Gutiérrez et al., 2025; Gutiérrez et al., 2025), iterative RAG

approaches rely on LLM-generated thoughts, making them prone to hallucination (Liu et al., 2024; Jiang et al., 2025). In contrast, GlobalRAG trained via reinforcement learning, strengthens global planning and coordinated reasoning, enabling more accurate evidence localization and integration in multi-hop QA.

Large Language Models and Reinforcement Learning. Reinforcement learning has markedly improved complex reasoning and self-verification in LLM post-training (OpenAI et al., 2024; Guo et al., 2025; Dong et al., 2025; Lai et al., 2025) and has rapidly extended to RAG (Jin et al., 2025a; Chen et al., 2025; Song et al., 2025a; Wang et al., 2025). Search-R1 (Jin et al., 2025a) introduces outcome-based rewards that incentivize more frequent invocations of the retriever. StepSearch (Wang et al., 2025) scores the per-step retrieval value using information gain and redundancy penalties to encourage informative query formulation. However, due to the lack of explicit global planning and the presence of unfaithful execution, these methods still underperform on multi-hop QA (Jin et al., 2025a; Wang et al., 2025). In contrast, we propose a Planning Quality Reward and a Subgoal Completion Reward to strengthen global reasoning in multi-hop QA.

3 Problem Formulation

We adopt the RL with Search Engine paradigm for multi-hop QA. Given a question q and corpus $\mathcal{C}$, the agent issues queries and receives retrieved sets over K hops, and a reader uses the collected evidence to produce the answer:

$$p(a \mid q, \mathcal{C}) \approx \sum_{r_{1:K},\, D_{1:K}} \left\{ \prod_{i=1}^{K} \Big[\pi_\theta\big(r_i \mid q, D_{<i}\big) \cdot p_{\text{se}}\big(D_i \mid r_i, \mathcal{C}\big) \Big] \right\} p_\phi\big(a \mid q, D_{1:K}\big). \tag{1}$$

Training objective.

$$\max_{\theta,\phi} \; \mathbb{E}_{(q,a^\star)\sim\mathcal{D}} \, \mathbb{E}_{\tau\sim\pi_\theta(\cdot|q,\mathcal{C})} \big[R(q,\tau,a^\star;\phi) \big] - \beta \, \mathrm{D}_{\mathrm{KL}} \Big[\pi_\theta(\cdot \mid q, \mathcal{C}) \,\Big\|\, \pi_{\text{ref}}(\cdot \mid q, \mathcal{C}) \Big]. \tag{2}$$

Here $p_{\text{se}}(\cdot \mid r_i, \mathcal{C})$ denotes the search-engine return distribution for query r_i, and $p_\phi(a \mid q, D_{1:K})$ denotes the reader's answer distribution given the gathered evidence. The reward $R(\cdot)$ is method-specific, and the KL term is optional for stabilizing policy updates.

4 Method

In this section, we present GlobalRAG, a framework for multi-hop reasoning. First, we introduce the rollout retrieval–enhanced GRPO algorithm. Second, we describe the generation of golden trajectories for reward supervision. Third, we detail the design of global planning–aware rewards. Finally, we present the progressive weight annealing strategy that integrates all rewards. An overview of the framework is illustrated in Figure 2.

4.1 Rollout Retrieval Enhanced GRPO for Multi-hop QA

An overview of the GRPO training scheme is shown in Figure 2(top). Formally, given an actor model π_θ that interacts with the retrieval engine to generate n rollouts for multi-hop QA, GRPO computes rewards from these rollouts and updates the actor model π_θ with reference to a frozen model π_{ref}. To adapt GRPO to our GlobalRAG, we follow Search-R1 (Jin et al., 2025a) and mask tokens between `<information>` and `</information>`, ensuring that the policy gradient is computed only on LLM-generated tokens. The overall training objective is then defined as:

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(|q)} \Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \min\left(\frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}\hat{A}_{i,t}, \; \text{clip}\left(\frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, 1-\varepsilon, 1+\varepsilon \right)\hat{A}_{i,t} \right) - \beta D_{\text{KL}}\left[\pi_\theta \,\|\, \pi_{\text{ref}}\right] \Bigg] \tag{3}$$

where $\hat{A}_{i,t} = \frac{R_{i,t} - \text{mean}(R_t)}{\text{std}(R_t)}$ is the normalized token-level advantage value for the t-th token of the i-th trajectory within the group; G is the group size; ε is the clipping ratio; β is the coefficient for KL divergence.

4.2 Golden Trajectory Generation with Global Planning

In this section, we introduce how to generate golden trajectories offline, as illustrated in Figure 2(bottom). The generation pipeline consists of three steps: *global planning*, *subgoal execution*, and *final answer generation*.

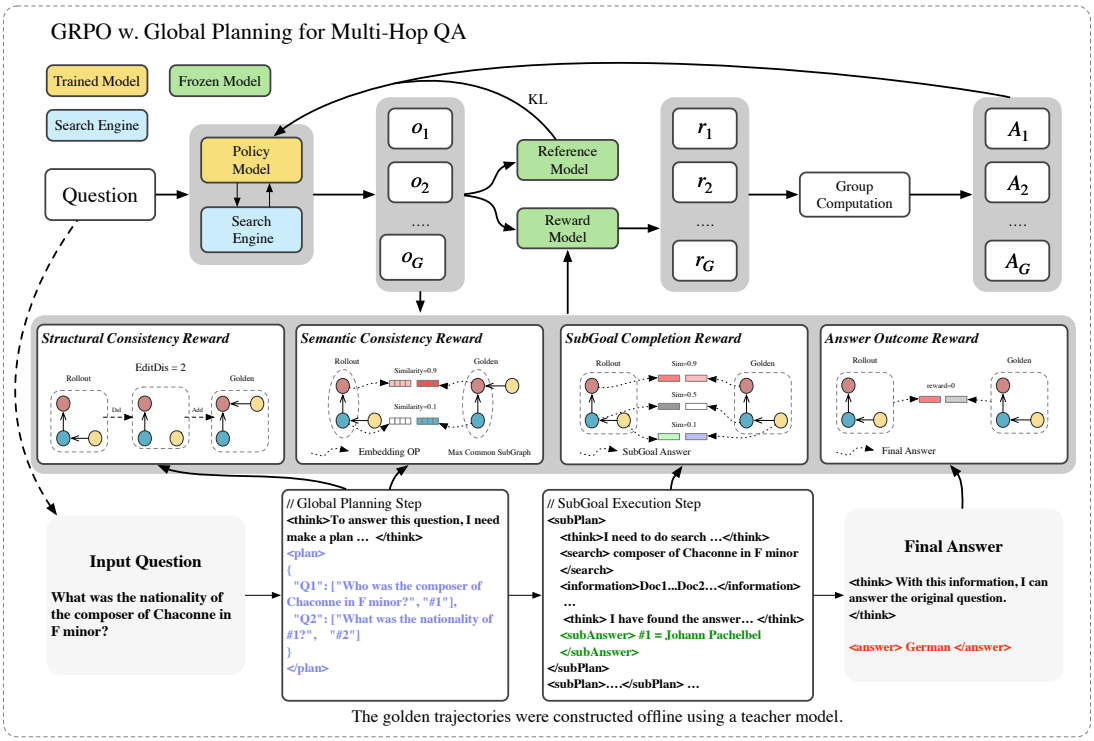

Figure 2: Overview of GlobalRAG. The top illustrates rollout retrieval–enhanced GRPO with a search engine; the bottom shows golden trajectories constructed by a teacher model; and the middle depicts the global planning–aware reward design used to update the policy model.

First, a teacher model performs global planning by analyzing the problem, decomposing it into sub-problems, and constructing a task-dependency graph with placeholder identifiers that define the execution paths. Second, the model solves sub-problems iteratively in topological order, conducting reasoning, retrieval, document acquisition, and sub-answer generation, while propagating intermediate results through placeholder substitution. Finally, the model produces the final answer by integrating the sub-answers according to the global plan and reasoning process.

4.3 Global Planning–Aware Reward Design

RL methods have shown great promise in RAG. However, relying solely on simple rule-based rewards, such as final-answer rewards or format rewards, often leads to shallow retrieval problems (Yu et al., 2025; Wang et al., 2025). In particular, for multi-hop QA without a global planning view, models tend to lose the original target, resulting in unfaithful execution. To address this issue, we design Global Planning–Aware Rewards as process-constrained signals to enhance the model's reasoning ability. Concretely, we introduce three rewards: *Structural Consistency Reward*, *Semantic Consistency Reward*, and *Subgoal Completion Reward*, as described below.

4.3.1 Planning Quality Reward

Our reward design evaluates the consistency between the rollout plan and the golden plan, both represented as directed acyclic graphs (DAGs) where nodes are subgoals and edges denote dependencies (Figure 2). We introduce two rewards: *Structural Consistency Reward*, measuring graph-level alignment, and *Semantic Consistency Reward*, assessing subgoal-level similarity. These rewards encourage more faithful and coherent reasoning.

Structural Consistency Reward enforces global planning awareness by evaluating the full dependency structure of complex tasks. In contrast to step-wise RL methods that optimize only the current step (Song et al., 2025b; He et al., 2025; Shi et al., 2025), we assess the entire graph topology to capture long-range dependencies and inter-task relationships. To quantify structural differences, we employ graph edit distance—the minimum number of operations required to transform G into $\tilde{G}$—and map it into a similarity score in $[0, 1]$ using an exponential function, with perfect alignment yielding the maximum reward.

Formally, given the rollout plan graph G and golden plan graph $\tilde{G}$, the structural consistency

reward is defined as:

$$R_{\text{str}} = \exp\big(-d_{\text{edit}}(G, \tilde{G})\big), \tag{4}$$

where $d_{\text{edit}}(G, \tilde{G})$ denotes the graph edit distance.

Semantic Consistency Reward evaluates the semantic alignment between the predicted plan and the golden plan. First, we identify the maximum common subgraph between the generated and reference plans to establish valid structural correspondences. Second, we compute semantic similarity only for matched subproblems within this subgraph. This design avoids spurious comparisons between structurally incompatible components and ensures evaluation focuses on semantically meaningful correspondences.

Formally, let $\mathcal{P}_{\text{gold}}$ denote the reference subproblems and $\mathcal{M}_P$ the set of matched subproblem pairs within the maximum common subgraph. For each matched pair i, $\rho^{(i)}_{\text{pred}}$ and $\rho^{(i)}_{\text{gold}}$ denote the embeddings of predicted and reference subproblems, respectively. The semantic consistency reward is defined as:

$$R_{\text{sem}} = \frac{1}{|\mathcal{P}_{\text{gold}}|} \sum_{i=1}^{|\mathcal{M}_P|} \cos\Big(\rho^{(i)}_{\text{pred}}, \rho^{(i)}_{\text{gold}}\Big), \tag{5}$$

where $\cos(\cdot,\cdot)$ denotes the cosine similarity.

4.3.2 Subgoal Completion Reward

This reward addresses the problem of unfaithful execution by encouraging the model to follow the global plan and produce reliable outputs for each subgoal. Unlike StepSearch(Wang et al., 2025), which applies rewards directly to the token sequence of generated search queries to stabilize training and increase signal density, we argue that this strategy overly constrains exploration and discourages the discovery of novel reasoning paths. Instead, our reward design focuses solely on the correctness of subgoal answers, without penalizing the search process, thereby leaving room for more exploratory and heuristic solution strategies.

Formally, for each matched subgoal $i \in \mathcal{M}_P$, let $o^{(i)}_{\text{pred}}$ and $o^{(i)}_{\text{gold}}$ denote predicted and gold answers, and $\phi(\cdot,\cdot)$ a similarity function. The reward is normalized by $|\mathcal{P}_{\text{gold}}|$ to ensure coverage, with unmatched nodes contributing zero:

$$R_{\text{step}} = \frac{1}{|\mathcal{P}_{\text{gold}}|} \sum_{i=1}^{|\mathcal{M}_P|} \phi\Big(o^{(i)}_{\text{pred}}, o^{(i)}_{\text{gold}}\Big). \tag{6}$$

4.3.3 Outcome Reward

A global plan and process rewards guide how the model reasons, but training must be closed by a terminal constraint requiring the correct final answer. The outcome reward enforces this by validating that execution yields the target solution, preventing credit on well-formed but unfaithful traces, and aligning optimization with end accuracy. We adopt a strict exact-match (EM) signal, applied after standard normalization, as a binary reward:

$$R_{\text{answ}} = \begin{cases} 1, & \text{if the answer exactly matches,} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

4.3.4 Format Reward

We treat the format reward as a front-door constraint that enforces a machine-parseable reasoning path aligned with our planning-centric RL pipeline, enabling the policy to learn *how* to plan, *what* to retrieve, and *where* to decide. This parseable protocol supports plan rewards, step supervision, stable policy-gradient training, and scalable evaluation.

The format reward is binary:

$$R_{\text{form}} = \begin{cases} 1, & \text{if the format is fully compliant,} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

4.3.5 Progressive weight annealing strategy

Inspired by prior work, we adopt a progressive annealing scheme of *structure first, accuracy later*. Early training emphasizes format, planning structure, semantics, and step execution, guiding the policy to construct and follow a coherent graph. Later training shifts weight toward the outcome signal to refine exactness without losing this discipline. The scheduling weight w_t balances process and outcome rewards:

$$w_t = \frac{1}{1 + \exp\left(\frac{t - 0.9T}{10}\right)}, \tag{9}$$

where t and T denote the current and total training steps.

The overall reward integrates all components with dynamic weighting:

$$R(t) = w_t\big(\alpha R_{\text{form}} + \lambda R_{\text{str}} + \gamma R_{\text{sem}} + \delta R_{\text{step}}\big) + R_{\text{answ}}, \tag{10}$$

with coefficients $\alpha = 0.1$ and $\lambda = \gamma = \delta = 0.5$. This curriculum smooths optimization, reduces

Methods	In-Domain QA						Out-of-Domain QA				Average	
	HotpotQA		2Wiki		MuSiQue		Bamboogle		WikiHop			
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Qwen2.5-3b-Base/Instruct												
Zero-Shot	1.2	3.3	1.6	2.7	0.0	1.2	0.0	1.9	0.1	0.7	0.6	2.0
NaiveRAG	6.5	13.8	2.2	4.1	0.5	1.9	4.0	8.5	2.8	7.6	3.2	7.2
IRCOT	4.8	11.2	3.0	8.6	0.9	3.2	5.6	10.1	2.8	6.3	3.4	7.9
Search-o1	14.6	22.1	14.0	17.0	3.6	7.7	26.4	34.9	5.3	10.3	12.8	18.4
Search-R1-base	26.8	36.7	25.6	29.2	2.5	7.3	11.2	20.4	9.3	16.5	15.1	22.0
Search-R1-instruct	31.3	41.5	31.3	36.5	7.7	13.2	28.0	34.7	8.2	13.8	21.3	27.9
StepSearch-base	29.1	40.3	33.5	38.7	10.4	17.7	32.8	41.8	**13.5**	19.6	23.9	31.6
StepSearch-instruct	28.9	39.9	31.9	38.3	9.5	16.6	32.0	43.8	11.2	18.1	22.7	31.3
GlobalRAG-base	30.4	41.0	**43.1**	**48.3**	**11.3**	17.4	**40.0**	**49.5**	12.7	19.7	**27.5**	35.2
GlobalRAG-instruct	**32.9**	**44.2**	42.3	47.8	10.8	**18.6**	37.6	49.3	12.9	**20.7**	27.3	**36.1**
Qwen2.5-7b-Base/Instruct												
Search-R1-base	28.5	38.3	26.7	31.3	7.1	12.9	35.2	43.1	11.4	17.4	21.8	28.6
Search-R1-instruct	35.9	47.5	33.4	39.5	9.7	16.7	43.4	52.7	15.5	23.0	27.6	35.9
StepSearch-base	33.5	43.5	39.1	44.8	13.1	22.1	42.4	52.4	16.9	24.6	29.0	37.5
StepSearch-instruct	34.8	46.9	36.3	42.5	13.4	21.4	41.6	53.1	14.9	22.3	28.2	37.2
GlobalRAG-base	33.0	44.4	47.0	52.4	14.2	**22.7**	46.4	51.0	16.0	22.7	31.3	38.6
GlobalRAG-instruct	**38.2**	**50.7**	**47.8**	**53.2**	**14.9**	21.2	**48.8**	**61.6**	**19.0**	**28.0**	**33.7**	**42.9**

Table 1: Multi-hop QA results (first three in-domain; last two out-of-domain). Best are **bold**. Methods without "base/instruct" default to instruct.

variance, coordinates retrieval under the global plan, and transitions from structural learning to end-to-end reasoning accuracy.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate on five multi-hop QA benchmarks: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), Bamboogle (Press et al., 2023), and WikiHop (Welbl et al., 2018), which require multi-step reasoning over evidence from multiple documents. We report results on the full test sets for Bamboogle and WikiHop, and on sampled subsets for HotpotQA, 2WikiMultiHopQA, and MuSiQue. For datasets with nonpublic test sets (e.g., HotpotQA, 2WikiMultiHopQA, MuSiQue, WikiHop), we use their development sets as test sets and report the corresponding results. Detailed statistics and corpus information are provided in Appendix C.1.

Baselines. To evaluate the effectiveness of GlobalRAG, we compare against representative baselines spanning prompting- and reinforcement-learning paradigms as well as static and dynamic retrieval strategies, organized into three categories: (1) **Inference without Retrieval**: Zero-Shot, where the model answers solely from parametric knowledge without external context. (2) **Retrieval-Augmented Inference**: NaiveRAG (Lewis et al., 2020), IRCOT (Trivedi et al., 2023), and Search-o1 (Li et al., 2025). (3) **Retrieval with RL**: Search-R1 (Jin et al., 2025a), StepSearch (Wang et al., 2025). We rely on authors' official implementations and released checkpoints where available to ensure comparability.

Evaluation Metrics. We adopt Exact Match (EM) and word-level F1 as the evaluation metrics. Formal definitions and normalization rules are provided in Appendix C.2.

Training and Implementation Details. We conduct experiments on two Qwen-2.5-3B (Base/Instruct) (Qwen et al., 2025). For retrieval, we build a `faiss-gpu` index over the 2018 Wikipedia snapshot (Karpukhin et al., 2020) and use E5 embeddings (Wang et al., 2022) as the dense retriever. To ensure fair comparison, we follow Search-R1(Jin et al., 2025a) and set the number of retrieved passages to 3 across all retrieval-based methods.

For training, we merge the training splits of HotpotQA, 2WikiMultiHopQA, and MuSiQue into a unified corpus and derive global-plan supervision

for GlobalRAG by inducing a subgoal–dependency plan and verifying intermediate steps, yielding 8,394 supervised trajectories.Our data generation is based on PER-PSE (Xu et al., 2025)

5.2 Main Results

Table 1 reports the comparison of GlobalRAG with competitive RL-trained and non-RL baselines across five datasets and summarizes the following observations:

(1) Overall SOTA with strong average gains: GlobalRAG achieves consistent improvements across all datasets. Specifically, we achieve the best results for evaluation metrics F1 on all datasets, and achieve four best results and one second-best result for EM. Relative to the strongest baseline (StepSearch), the average improvement in F1 is 14.2%: HotpotQA +9.7%, 2Wiki +23.5%, MuSiQue +5.1%, Bamboogle +17.9%, and WikiHop +5.6%. These results validate the effectiveness of our planning-aware reinforcement-learning framework for strengthening global reasoning in multi-hop QA.

(2) Robust in-domain and out-of-domain performance: GlobalRAG achieves leading performance on both in-domain datasets (HotpotQA, 2Wiki, and MuSiQue) and the out-of-domain datasets (Bamboogle and WikiHop). Concretely, it delivers top results on the in-domain benchmarks (HotpotQA, 2Wiki, MuSiQue) and maintains strong advantages on the out-of-domain benchmarks (Bamboogle, WikiHop), indicating strong generalization to unseen corpora. In comparison, Search-R1 performs well on HotpotQA but degrade notably out of domain, suggesting that generalization remains challenging for existing methods.

(3) Data efficiency and training scale: Despite using far less data, GlobalRAG achieves robust performance. Trained on only 8k examples (approximately 42% of StepSearch's 19k and 4.7% of Search-R1's 170k; see Table 2 for details), it attains average relative improvements of 14.2% EM and 14.2% F1 over StepSearch, and 28.2% EM and 29.4% F1 over Search-R1. We attribute GlobalRAG's sample efficiency to two complementary design choices that directly address two limitations: the absence of global planning and unfaithful execution. First, planning-aware process supervision (via Planning Quality Reward and SubGoal Completion Reward) provides dense, graph-aligned and semantically aligned signals that guide exploration and reduce reliance on large data budgets. Second, progressive weight annealing gradually rebalances process-oriented and outcome-oriented objectives in the later stages of training, thereby reducing overfitting and improving generalization.

Methods	**Search-R1**	**StepSearch**	**GlobalRAG**
Train Size	170k	19k	8k

Table 2: Comparison of training data sizes for different methods.

Methods	**HotpotQA**		**2Wiki**		**MuSiQue**		**Bamboogle**		**WikiHop**	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
w/o R_{sem}	30.0	41.5	38.1	44.1	10.1	17.2	37.6	51.4	10.4	18.1
w/o R_{str}	30.6	42.2	39.6	46.0	9.5	18.3	39.2	47.9	11.2	18.7
w/o R_{step}	27.6	39.2	34.7	41.4	8.8	15.3	36.0	47.9	9.0	16.8
w/o w_t	30.6	42.6	39.7	45.9	10.4	18.1	33.6	46.5	12.4	20.1
GlobalRAG	**32.9**	**44.2**	**42.3**	**47.8**	**10.8**	**18.6**	**37.6**	**49.3**	**12.9**	**20.7**

Table 3: Ablation study on reward components.

(4) Generalization across backbones and model sizes: Across Qwen2.5 backbones—covering both 3B and 7B capacities and both *Base* and *Instruct* variants—scaling the backbone uniformly improves all methods (Table 1). Crucially, GlobalRAG remains the top performer under every backbone setting, and its margin is preserved (often widened) when moving from 3B to 7B. These results indicate that our planning-aware RL is *backbone-agnostic* (robust to the choice between Base/Instruct) and *capacity-compatible* (benefiting from larger models), demonstrating strong plug-and-play generalization across different foundation models.

5.3 Ablation Study

To quantify the effectiveness of the plan-to-execution reward design, we conduct a systematic ablation on GlobalRAG, and report the results in Table 3.

To validate global planning, we ablate the two plan-level rewards and evaluate them jointly. *w/o* R_{str} removes supervision over the global plan graph, and *w/o* R_{sem} removes supervision over the semantic composition of subgoals. Across all datasets, both settings reduce EM/F1, with the degradation more pronounced for R_{sem} than for R_{str}. These results show that plan-level guidance is essential beyond R_{step} supervision: the structural signal controls long-range dependencies and cross-hop coordination, while the semantic signal anchors faithful decomposition that reliably steers retrieval and reasoning. Together they form the

core of global planning in GlobalRAG.

To assess intermediate execution, we introduce GlobalRAG *w/o* R_{step}, which removes supervision on the correctness of intermediate subgoal execution. This variant yields the largest declines in Table 3; averaged over five datasets, EM decreases by 17.7% and F1 by 12.5%. The magnitude and consistency of these drops indicate that constraining the search module alone is insufficient, and that validating intermediate execution via R_{step} is necessary to prevent partial subgoal answers from being promoted to final predictions and to ensure faithful plan execution.

We also disable progressive reward-weight annealing by fixing reward weights during training (i.e., w_t is held constant for all t); we denote this variant as w/o w_t. A consistent decline indicates that while the w_t-based curriculum improves stability, it is not the primary contributor to the gains.

5.4 Retrieval Performance for Multi-hop QA

To explain why GlobalRAG attains stronger multi-hop QA results, we analyze its retrieval behavior. As shown in Figure 3, both GlobalRAG and StepSearch typically conduct about twice as many retrievals as Search-R1 (on WikiHop, about 1.2× as many as Search-R1). These confirm that, compared with relying solely on outcome-based rewards, incorporating process-oriented rewards more effectively promotes iterative retrieval, thereby enabling the model to identify the correct answer.

Importantly, GlobalRAG uses comparable but on average slightly fewer retrievals than StepSearch (Figure 3), yet it attains better EM and F1 (Table 1). This indicates that adding planning quality reward and subgoal completion reward effectively enhances global planning and faithful execution, thereby achieving higher answer accuracy without increasing the number of retrieval steps.

5.5 Case Study

The case study in Table 4 shows that GlobalRAG first decomposes complex multi-hop questions into structured sub-questions with explicit dependencies (Q1→Q2), then faithfully executes this plan via consistent reasoning and placeholder-grounded integration (1 = Parramatta River, 2 = Sydney Harbour). This systematic procedure reduces the entity confusions prevalent in baselines and clarifies how intermediate evidence flows into the final answer.

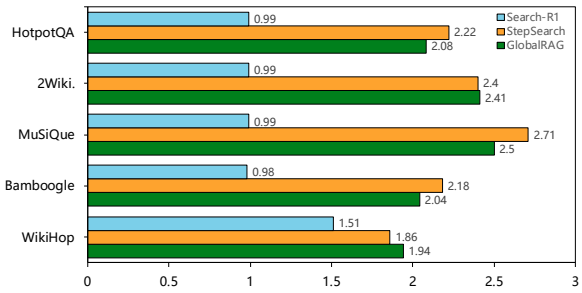

Figure 3: Comparison of average retrieval counts across datasets for Search-R1, StepSearch and GlobalRAG.

Question: Into what does the river for which Toongabbie Creek serves as the mouth flow?
Answer: Sydney Harbour

<think> To answer this question, I can proceed step-by-step to find the answer. </think>

<plan>
{
"Q1": ["For what river does Toongabbie Creek serve as the mouth?", "#1"],
"Q2": ["Into what does #1 flow?", "#2"]
}
</plan>

<subPlan>
<think> First, I need to identify the river that Toongabbie Creek serves as the mouth of. </think>
<search> river for which Toongabbie Creek serves as the mouth </search>
<information> ... The creek flows ... before **reaching its confluence with the Darling Mills Creek to form the Parramatta River** ... </information>
<think>Toongabbie Creek flows into the Parramatta River. </think>
<subAnswer> #1 = Parramatta River </subAnswer>
</subPlan>

<subPlan>
<think> Next, I need to determine what the Parramatta River flows into. </think>
<search> Parramatta River flows into </search>
<information> ... Parramatta River ... **The Parramatta River is the main tributary of Sydney Harbour, a branch of Port Jackson** ... Here it **flows into Port Jackson** ... </information>
<think> The Parramatta River flows into Sydney Harbour. </think>
<subAnswer> #2 = Sydney Harbour </subAnswer>
</subPlan>

<think> With this information, I can provide the final answer to the question. </think>

<answer> **Sydney Harbour** </answer>

Table 4: Detailed trace of GlobalRAG reasoning process for a multi-hop question.

Additional illustrative cases and parameter settings are provided in Appendix D.

6 Conclusions

In this work, we introduced GlobalRAG, a reinforcement learning framework that enhances multi-hop QA by integrating global planning and faithful execution. Through the use of planning-aware rewards and progressive weight annealing, GlobalRAG effectively aligns subgoal decomposition with evidence retrieval and reasoning. Experimental results on both in-domain and out-of-domain datasets validate its robustness and generalization ability, showing consistent improvements over strong baselines. These findings highlight the importance of plan-centric optimization in bridging retrieval and reasoning, and point toward broader applications of reinforcement learning in complex reasoning tasks.

7 Limitations

Our study has three main limitations. First, due to computational and cost constraints, we are unable to conduct RL training on very large-scale models (e.g., DeepSeek-R1). Whether our approach can effectively transfer to such models remains an open question. Second, our method tends to produce longer chains of thought, which increases token consumption and inference latency. Third, our work primarily focuses on multi-hop QA, without detailed analysis or training data generation for single-hop tasks.

References

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. 2025. Research: Learning to reason with search for llms via reinforcement learning. *Preprint*, arXiv:2503.19470.

Rong Cheng, Jinyi Liu, Yan Zheng, Fei Ni, Jiazhen Du, Hangyu Mao, Fuzheng Zhang, Bo Wang, and Jianye Hao. 2025. Dualrag: A dual-process approach to integrate reasoning and retrieval for multi-hop question answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 2025. Agentic reinforced policy optimization. *Preprint*, arXiv:2507.19849.

Jinyuan Fang, Zaiqiao Meng, and Craig Macdonald. 2025. Kirag: Knowledge-driven iterative retriever for enhancing retrieval-augmented generation. *Preprint*, arXiv:2502.18397.

Zhangyin Feng, Xiaocheng Feng, Dezhi Zhao, Maojin Yang, and Bing Qin. 2024. Retrieval-generation synergy augmented large language models. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11661–11665.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2025. Hipporag: neurobiologically inspired long-term memory for large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From RAG to memory: Non-parametric continual learning for large language models. In *Forty-second International Conference on Machine Learning*.

Jie He and 1 others. 2025. From sufficiency to reflection: Reinforcement-guided thinking quality in retrieval-augmented reasoning for llms. *arXiv preprint arXiv:2507.22716*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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).

Zhouyu Jiang, Mengshu Sun, Lei Liang, and Zhiqiang Zhang. 2025. Retrieve, summarize, plan: Advancing multi-hop question answering with an iterative approach. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 1677–1686, New York, NY, USA. Association for Computing Machinery.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025a. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*.

Jiajie Jin, Yutao Zhu, Zhicheng Dou, Guanting Dong, Xinyu Yang, Chenghao Zhang, Tong Zhao, Zhao Yang, and Ji-Rong Wen. 2025b. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 737–740.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Hanyu Lai, Xiao Liu, Yanxiao Zhao, Han Xu, Hanchen Zhang, Bohao Jing, Yanyu Ren, Shuntian Yao, Yuxiao Dong, and Jie Tang. 2025. Computerrl: Scaling end-to-end online reinforcement learning for computer use agents. *Preprint*, arXiv:2508.14040.

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 *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Ruosen Li and Xinya Du. 2023. Leveraging structured information for explainable multi-hop question answering and reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6779–6789, Singapore. Association for Computational Linguistics.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*.

Yanming Liu, Xinyue Peng, Xuhong Zhang, Weihao Liu, Jianwei Yin, Jiannan Cao, and Tianyu Du. 2024. RA-ISF: Learning to answer and understand from retrieval augmentation via iterative self-feedback. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4730–4749, Bangkok, Thailand. Association for Computational Linguistics.

Keer Lu, Chong Chen, Bin Cui, Huang Leng, and Wentao Zhang. 2025. Pilotrl: Training language model agents via global planning-guided progressive reinforcement learning. *Preprint*, arXiv:2508.00344.

Lang Mei, Zhihan Yang, and Chong Chen. 2025. Ai-searchplanner: Modular agentic search via pareto-optimal multi-objective reinforcement learning. *Preprint*, arXiv:2508.20368.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 244 others. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

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

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, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274, Singapore. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Yaorui Shi, Sihang Li, Chang Wu, Zhiyuan Liu, Junfeng Fang, Hengxing Cai, An Zhang, and Xiang Wang. 2025. Search and refine during think: Facilitating knowledge refinement for improved retrieval-augmented reasoning. *Preprint*, arXiv:2505.11277.

Zhengliang Shi, Shuo Zhang, Weiwei Sun, Shen Gao, Pengjie Ren, Zhumin Chen, and Zhaochun Ren. 2024. Generate-then-ground in retrieval-augmented generation for multi-hop question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7339–7353, Bangkok, Thailand. Association for Computational Linguistics.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025a. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *CoRR*, abs/2503.05592.

Huatong Song and 1 others. 2025b. R1-searcher++: Scaling up llms as effective web searchers via reinforcement learning. *arXiv preprint arXiv:2505.17005*.

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.

Hongxin Wang, Zheng He, Jing Chen, and 1 others. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Ziliang Wang, Xuhui Zheng, Kang An, Cijun Ouyang, Jialu Cai, Yuhang Wang, and Yichao Wu. 2025. Stepsearch: Igniting llms search ability via step-wise proximal policy optimization. *Preprint*, arXiv:2505.15107.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Sheng Xu, Diyi Yang, and 1 others. 2025. Beyond the answer: Advancing multi-hop qa with fine-grained graph reasoning and evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

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.

Chuanyue Yu, Kuo Zhao, Yuhan Li, Heng Chang, Mingjian Feng, Xiangzhe Jiang, Yufei Sun, Jia Li, Yuzhi Zhang, Jianxin Li, and Ziwei Zhang. 2025. Graphrag-r1: Graph retrieval-augmented generation with process-constrained reinforcement learning. *CoRR*, abs/2507.23581.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Liu Yong, and Shen Huang. 2024. End-to-end beam retrieval for multi-hop question answering. 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 1718–1731, Mexico City, Mexico. Association for Computational Linguistics.

Qi Zhang, Shouqing Yang, Lirong Gao, Hao Chen, Xiaomeng Hu, Jinglei Chen, Jiexiang Wang, Sheng Guo, Bo Zheng, Haobo Wang, and Junbo Zhao. 2025. Lets: Learning to think-and-search via process-and-outcome reward hybridization. *Preprint*, arXiv:2505.17447.

Shu Zhao, Tan Yu, Anbang Xu, Japinder Singh, Aaditya Shukla, and Rama Akkiraju. 2025. Parallelsearch: Train your llms to decompose query and search subqueries in parallel with reinforcement learning. *arXiv preprint arXiv:2508.09303*.

Rongzhi Zhu, Xiangyu Liu, Zequn Sun, Yiwei Wang, and Wei Hu. 2025. Mitigating lost-in-retrieval problems in retrieval augmented multi-hop question answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22362–22375, Vienna, Austria. Association for Computational Linguistics.

A Prompt for Rollout

The prompt we use for trajectory generation is shown in Figure 4.

Prompt Used for Trajectory Generation

Answer the given question by following the steps below. You must conduct all reasoning inside <think> and </think> before producing <plan>, <search>, <subAnswer> and <answer>.

Step 1: Explicitly generate one or more sub-questions within the <plan> and </plan> block.

- Each sub-question must contain both a question and a placeholder (#1, #2, etc.) that represents the answer to that question.
- Each sub-question should be as brief and precise as possible.
- If a sub-question depends on the answer to a previous one, use a placeholder (#1, #2, etc.) to represent that dependency.

Step 2: For each sub-question, create a block enclosed in <subPlan> and </subPlan>.

- In sequential order, take one sub-question from <plan> and fill it between <search> and </search>.
- Conclude the block with a <subAnswer> that binds the answer to the current sub-question.

Step 3: Provide the final result inside <answer> and </answer>, without detailed explanations.

Now, it's your turn! Please answer the following question!!!

Question: {Question}

Figure 4: The prompt used for trajectory generation

B Analysis of RL-based RAG Problems

We sampled 300 instances from each of five public multi-hop reasoning benchmarks—HotpotQA, 2WikiMultiHopQA (2Wiki), MuSiQue, Bamboogle, and WikiHop. To identify primary failure modes, we ran Search-R1 on each 300-item subset and isolated error cases: 219 (HotpotQA), 214 (2Wiki), 275 (MuSiQue), 90 (Bamboogle), and 274 (WikiHop). For each error, we then assessed whether the gold answer appeared in any document retrieved across multiple search rounds. If the answer never appeared, we labeled the case as a failure to retrieve answer-bearing evidence. The prevalence of this retrieval-failure criterion was 58.90% (HotpotQA), 44.39% (2Wiki), 85.45% (MuSiQue), 76.67% (Bamboogle), and 85.04% (WikiHop).

We further analyzed the reasoning traces of the "no-answer-retrieved" subset using DeepSeek-R1 and identified two dominant error categories: (i) absence of global planning, where the model fails to construct an overall plan for decomposing and solving the question; and (ii) unfaithful execution, where retrieval drifts from the original target and results in missing the correct document. The distributions were as follows—HotpotQA: global-planning absence, 63.57%; unfaithful execution, 36.43%; other, 0.00%. 2Wiki: 50.53% vs. 47.37% vs. 2.10%; MuSiQue: 67.23% vs. 30.64% vs. 2.13%; Bamboogle: 56.52% vs. 37.68% vs. 5.80%; WikiHop: 56.57% vs. 41.97% vs. 1.46%. These results indicate that, when answer-bearing contexts are not retrieved, the absence of global planning and unfaithful execution constitute the predominant sources of failure.

C Experimental Details

C.1 Datasets

Our experiments utilize five multi-hop QA datasets: HotPotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), Bamboogle (Press et al., 2023) and WikiHop (Welbl et al., 2018). For each dataset, we construct the retrieval corpus following the standardized procedure established in prior (Jin et al., 2025b). The entire system is deployed as an online retrieval service using FastAPI for efficient query processing.

The training set for GlobalRAG is constructed from the train splits of HotpotQA 2WikiMultiHopQA and MuSiQue, totaling 8,394 examples. Moreover, we precompute golden plan and golden graph for all the data in the training datasets. HotpotQA, 2WikiMultihopQA, MuSiQue and Bamboogle are sourced from the FlashRAG Datasets

collection. Since the test or development sets of HotpotQA, 2WikiMultihopQA, and MuSiQue partially overlap with our training data, we perform deduplication. After processing, 5,187 samples are retained from the HotpotQA development set, 8,372 from the 2WikiMultihopQA test set, 444 from the MuSiQue development set. For Bamboogle, the entire datasets are directly used as test sets, including 125 samples, respectively. For WikiHop, the entire dev sets are used as test sets, including 5,129 samples.

C.2 Metrics

Following the settings in DualRAG (Cheng et al., 2025), we adopt the following evaluation metrics:

1. Exact Match (EM)

The EM metric measures whether the predicted answer is exactly the same as the ground truth:

$$\text{EM} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(\hat{y}_i = y_i) \tag{11}$$

2. F1 Score

The F1 score is the harmonic mean of token-level precision and recall computed between the predicted and gold token sets, with values in the range $[0, 1]$:

$$F1(y, \hat{y}) = \frac{2 \times P \times R}{P + R} \tag{12}$$

where Precision (P) and Recall (R) are defined as:

$$P = \frac{|\text{CommonTokens}(y, \hat{y})|}{|\text{Tokens}(\hat{y})|}, \tag{13}$$

$$R = \frac{|\text{CommonTokens}(y, \hat{y})|}{|\text{Tokens}(y)|} \tag{14}$$

Precision and recall are also commonly used for evaluation in QA tasks.

C.3 Training Hyperparameters

We trained our model on $4\times$ A800 GPUs using a batch size of 256 and sequence length of 4,096 tokens, with FSDP and CPU offloading for memory optimization. The training employed GRPO with $\beta = 0.001$, $\epsilon = 0.2$, learning rate of 1×10^{-6}, and standard sampling parameters (temperature = 1.0, top-p = 1.0). We generated 5 responses per prompt over 200 steps, saving checkpoints every 10 steps, with an action budget of 4 and top-3 paragraph retrieval.

D Case Study

To better understand the performance characteristics of our proposed method, we conduct a comparative analysis between GlobalRAG and Search-R1 through case studies. Our analysis reveals that GlobalRAG demonstrates significant advantages over the baseline approach in two critical aspects: enhanced retrieval precision through global planning and improved reasoning accuracy via structured sub-question decomposition.

Specifically, the global planning strategy employed by our method enables more precise information retrieval and acquisition of accurate relevant information. As demonstrated in Table 5 and Table 6, GlobalRAG successfully identifies and retrieves the most pertinent information needed for answering complex queries, whereas Search-R1 often retrieves incomplete or less relevant content. Furthermore, the explicit incorporation of `<subPlan>` steps for structured reasoning, combined with a reward mechanism for validating sub-question answers, allows our method to effectively filter out distracting information while ensuring the correctness of both intermediate reasoning processes and sub-answers. This advantage is particularly evident in Table 7 and Table 8, where GlobalRAG maintains reasoning coherence and accuracy despite the presence of potentially misleading information. These findings demonstrate that our method's architectural innovations contribute to enhanced accuracy and reliability in complex question-answering tasks.

Case 1: Into what does the river for which Toongabbie Creek serves as the mouth flow?
Ground Truth: Sydney Harbour

GlobalRAG:

<think> To answer this question, I need to break it down into sub-questions. First, I need to identify the river that Toongabbie Creek flows into. Then, I need to determine what that river flows into. I can proceed step-by-step to find the answer. </think>

<plan> { "Q1": ["For what river does Toongabbie Creek serve as the mouth?", "#1"], "Q2": ["Into what does #1 flow?", "#2"] } </plan>

<subPlan>
<think> First, I need to identify the river that Toongabbie Creek serves as the mouth of. </think>
<search> river for which Toongabbie Creek serves as the mouth </search>
<information> Doc 1: ... The creek flows generally south-west, then south by east, then south by west, then east before reaching its **confluence with the Darling Mills Creek to form the Parramatta River**, in the suburb of ... </information>
<think> The information provided indicates that Toongabbie Creek flows into the Parramatta River. </think>
<subAnswer> **#1 = Parramatta River** </subAnswer>
</subPlan>

<subPlan>
<think> Next, I need to determine what the Parramatta River flows into. </think>
<search> Parramatta River flows into </search>
<information> Doc 1: Parramatta River The Parramatta River is an intermediate tide dominated, drowned valley estuary located in Sydney, New South Wales, Australia. **The Parramatta River is the main tributary of Sydney Harbour**, a branch of Port Jackson ... the river flows in an easterly direction to a line between Yurulbin, Birchgrove and Manns Point, Greenwich. Here it **flows into Port Jackson** ... </information>
<think> The information provided indicates that the Parramatta River flows into Sydney Harbour. </think>
<subAnswer> **#2 = Sydney Harbour** </subAnswer>
</subPlan>

<think> To answer the original question, I need to combine the answers from the sub-questions. With this information, I can determine into what river Toongabbie Creek flows. With this, I can provide the final answer to the question. </think>

<answer> **Sydney Harbour** </answer>

Search-R1:

<think> To find out what river the Toongabbie Creek serves as the mouth to, I will conduct a search for this information. </think>

<search> Into what does the river for which Toongabbie Creek serves as the mouth flow </search>

<information> Doc 1: Toongabbie Creek, an urban watercourse that is part of the Parramatta River catchment, is located in Greater Western Sydney, New South Wales, Australia ... The creek flows generally south-west, then south by east, then south by west, then east before reaching its **confluence with the Darling Mills Creek to form the Parramatta River**, in the suburb of ... </information>

<think> Based on the information retrieved: </think>

<search> what river does the mouth of Toongabbie Creek flow into </search>

<information> ... </information>

<think> From the provided documents: </think>

<answer> **Parramatta River** </answer>

Table 5: Case Study 1. The predictions are colored **Red** if the model answers incorrectly and **Green** if it answers correctly. The core evidences are colored **Blue**.

Case 2: Who was the first African American mayor of the most populous city in the United States?
Ground Truth: David Dinkins

GlobalRAG:

<think> To answer this question, I need to break it down into sub-questions. First, I need to identify the most populous city in the United States. Then, I need to find the first African American mayor of that city. I can reason through the sub-questions to get to the final answer. </think>

<plan> { "Q1": ["What is the most populous city in the United States?", "#1"], "Q2": ["Who was the first African American mayor of #1?", "#2"] } </plan>

<subPlan>
<think> First, I need to determine the most populous city in the United States. </think>
<search> most populous city in the United States </search>
<information> Doc 1: New York City The City of New York, often called New York City (NYC) or simply New York (NY), is **the most populous city in the United States** ... </information>
<think> The information provided indicates that the most populous city in the United States is New York City. </think>
<subAnswer> **#1 = New York City** </subAnswer>
</subPlan>

<subPlan>
<think> Next, I need to find the first African American mayor of New York City. </think>
<search> first African American mayor of New York City </search>
<information> ... David Dinkins David Norman Dinkins (born July 10, 1927) is an American politician, lawyer, and author who served as the 106th Mayor of New York City, from 1990 to 1993. He was **the first and, to date, the only African American to hold that office** ... </information>
<think> The information provided indicates that the first African American mayor of New York City was David Dinkins. </think>
<subAnswer> **#2 = David Dinkins** </subAnswer>
</subPlan>

<think> Now that I have the necessary subAnswers, I can determine the first African American mayor of the most populous city in the United States. With this information, I will be able to provide the final answer to the question. </think>

<answer> **David Dinkins** </answer>

Search-R1:

<think> To find out who the first African American mayor of the most populous city in the United States was, I will conduct some research. </think>

<search> first African American mayor of the most populous city in the United States </search>

<information> Doc 1: ... she became the **first female African American Mayor of a major metropolitan city** [Compton, California] ... Doc 2: Robert Clayton Henry ... was Mayor of Springfield, Ohio from 1966 to 1968. He was the **first African-American mayor of an American city of any size** ... Doc 3: Theodore M. Berry ... was the **first African-American mayor of Cincinnati, Ohio** ... </information>

<think> Based on the research: </think>

<search> who was the first African American mayor of the most populous city in the United States </search>

<information> ... </information>

<think> From the data, I realize Robert C. Henry could be wrongly aligned, Ted Berry indeed form Document important positions. </think>

<answer> **Ted Berry** </answer>

Table 6: Case Study 2. The predictions are colored **Red** if the model answers incorrectly and **Green** if it answers correctly. The core evidences are colored **Blue**.

Case 3: Who is the spouse of the producer of *Dolores Claiborne*?
Ground Truth: Helen Mirren

GlobalRAG:

<think> To answer this question, I need to break it down into sub-questions. First, I need to identify the producer of the film Dolores Claiborne. Then, I need to find who that producer is married to. I can proceed step-by-step to find the answer. </think>

<plan> { "Q1": ["Who is the producer of Dolores Claiborne?", "#1"], "Q2": ["Who is the spouse of #1?", "#2"] } </plan>

<subPlan>
<think> First, I need to identify the producer of the film Dolores Claiborne. </think>
<search> producer of Dolores Claiborne film </search>
<information> Doc 1: **Dolores Claiborne is a 1995 American psychological thriller drama film directed by Taylor Hackford** and starring Kathy Bates, Jennifer Jason Leigh, and David Strathairn ... </information>
<think> The information clearly states that the film was directed by Taylor Hackford. </think>
<subAnswer> **#1 = Taylor Hackford** </subAnswer>
</subPlan>

<subPlan>
<think> Next, I need to find who Taylor Hackford is married to. </think>
<search> Taylor Hackford spouse </search>
<information> Doc 1: ... Hackford has been married three times ... Doc 2: ... **Hackford has been married to Academy Award-winning actress Helen Mirren since 1997** ... </information>
<think> The information indicates that Taylor Hackford is married to Helen Mirren. </think>
<subAnswer> **#2 = Helen Mirren** </subAnswer>
</subPlan>

<think> Now that I have the necessary sub Answers, I can determine the spouse of the producer of Dolores Claiborne. </think>

<answer> **Helen Mirren** </answer>

Search-R1:

<think> To find the spouse of the producer of "Dolores Claiborne", I need to gather details on the producer of the film. I will conduct a search to locate that information. </think>

<search> producer of Dolores Claiborne </search>

<information> Doc 1: Dolores Claiborne is a 1995 American psychological thriller drama film directed by Taylor Hackford ... **The screenplay for "Dolores Claiborne" was adapted by Tony Gilroy** ... </information>

<think> From the findings, I confirm that the producer of "Dolores Claiborne" is Tony Gilroy. </think>

<search> who is the spouse of Tony Gilroy </search>

<information> Doc 1: Tony Gilroy ... He has two children, Sam and Kathryn, and is married to **Susan Gilroy** ... </information>

<think> From this information: </think>

<answer> **Susan Gilroy** </answer>

Table 7: Case Study 3. The predictions are colored **Red** if the model answers incorrectly and **Green** if it answers correctly. The core evidences are colored **Blue**.

Case 4: When was the Western Germanic language spoken from which the small settlement situated on the river Leda opposite Leer derives its name?
Ground Truth: between the 8th and 16th centuries

GlobalRAG:

<think> To answer this question, I need to break it down into sub-questions. First, ... Then, ... Finally, ... I will follow the steps to generate sub-questions and answer them sequentially. </think>

<plan> { "Q1": ["Which small settlement derives its name from the river Leda?", "#1"], "Q2": ["From which Western Germanic language does #1 derive its name?", "#2"], "Q3": ["When was the Western Germanic language spoken from which #1 derives its name?", "#3"] } </plan>

<subPlan>
<think> First, I need to identify the small settlement that derives its name from the river Leda. </think>
<search> small settlement named after river Leda opposite Leer </search>
<information> Doc 3: ... On the southern bank of the Leda, in the "Overledingen Land" (Overledingen="country over the Leda"), opposite Leer, lies the **small settlement of Kloster Muhde** ("Muhde" from the **Old Frisian "mutha"** meaning "(river) mouth") ... </information>
<think> The information provided indicates that the small settlement situated on the river Leda opposite Leer is Kloster Muhde. </think>
<subAnswer> **#1 = Kloster Muhde** </subAnswer>
</subPlan>

<subPlan>
<think> Next, I need to determine the Western Germanic language from which the name of Kloster Muhde derives. </think>
<search> origin of name of Kloster Muhde in Lower Saxony </search>
<information> ... </information>
<think> ... Western Germanic language from which the name derives is Old Frisian. </think>
<subAnswer> **#2 = Old Frisian** </subAnswer>
</subPlan>

<subPlan>
<think> Finally, I need to find out when the Western Germanic language from which Kloster Muhde derives its name was spoken. </think>
<search> when Old Frisian language was spoken </search>
<information> Doc 1: Old Frisian Old Frisian is a West Germanic language **spoken between the 8th and 16th centuries** in the area ... </information>
<think> The information indicates that the Old Frisian language, from which the name of Kloster Muhde derives, was spoken between the 8th and 16th centuries. </think>
<subAnswer> **#3 = between the 8th and 16th centuries** </subAnswer>
</subPlan>

<think> ... With this information, I can provide the final answer to the question. </think>

<answer> **between the 8th and 16th centuries** </answer>

Search-R1:

<think> To find out ... , I need to first identify the settlement's name and the historical context surrounding its name. </think>

<search> name of settlement on river Leda opposite Leer Western Germanic language </search>

<information> ... (Overledingen="country over the Leda"), opposite Leer, lies the **small settlement of Kloster Muhde** ("Muhde" from the **Old Frisian "mutha"** meaning ... </information>

<think> From the information retrieved: The small settlement on the river Leda opposite Leer is named Kloster Muhde. </think>

<search> when was Western Germanic language spoken </search>

<information> ... Germanic dialects were closely enough related to have been **mutually intelligible up to the 7th century**... successively. The High German consonant shift that occurred mostly during the **7th century AD** in what is now southern Germany, Austria, and Switzerland can be considered **the end of the linguistic unity among the West Germanic dialects** ... </information>

<think> Based on these documents: </think>

<answer> **7th century** </answer>

Table 8: Case Study 4. The predictions are colored **Red** if the model answers incorrectly and **Green** if it answers correctly. The core evidences are colored **Blue**.