# 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 constrained by two fundamental issues: (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.

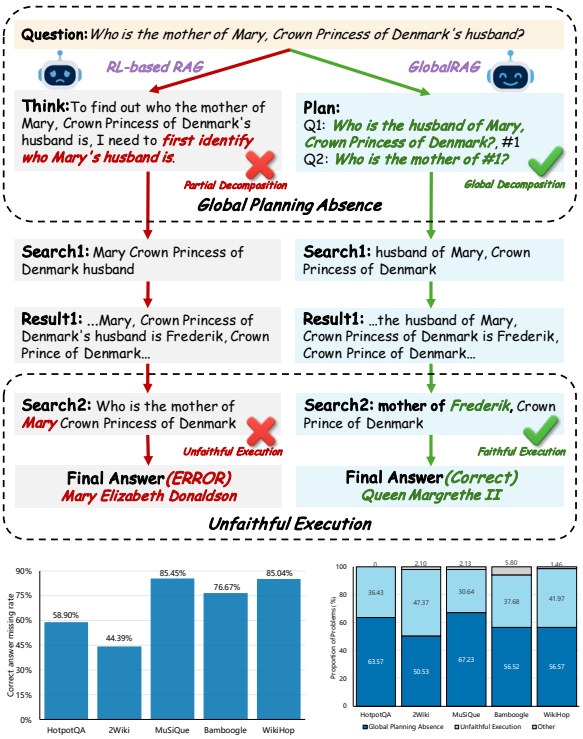




Figure 1: (*top*) Example of global planning absence and unfaithful execution. (*bottom left*) The most frequent failure of Search-R1 is missing the correct document. (*bottom right*) Error distribution where these two issues account for over 94% of failures.

## 1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has recently demonstrated strong empirical gains (Guo et al., 2025; Dong et al., 2025; Lai et al., 2025), particularly in retrieval-augmented generation (RAG), where RL-based methods achieve competitive performance by explicitly optimizing reasoning and retrieval behaviors (Jin et al., 2025a; Chen et al., 2025; Song et al., 2025a). Despite this progress, existing approaches still struggle with multi-hop and compositional questions, where reasoning requires long-horizon planning and consistent execution across multiple retrieval steps.

Recent studies attempt to mitigate this limitation by encouraging more frequent or structured retrieval, such as *StepSearch* (Wang et al., 2025), *LeTS* (Zhang et al., 2025), *ParallelSearch* (Zhao et al., 2025), and *GraphRAG-R1* (Yu et al., 2025). While these methods alleviate shallow retrieval to some extent, they do not fundamentally address the dominant failure modes in multi-hop QA.

Through an empirical analysis on five representative multi-hop benchmarks—HotpotQA, 2WikiMultiHopQA, MuSiQue, Bamboogle, and WikiHop—we find that the majority of errors arise from the model's inability to retrieve documents containing the correct answer after multiple retrieval steps. A finer-grained inspection of reasoning trajectories further reveals two dominant causes: **global plan-**

**ning absence**, where the model fails to construct a coherent decomposition of the question, and **unfaithful execution**, where retrieval and reasoning drift away from the intended subgoals. Together, these two issues account for over 94% of failures in cases where the correct answer is missing (details in Appendix B).

Motivated by these observations, we propose **GlobalRAG**, a reinforcement learning framework designed to enhance global reasoning in multi-hop QA. To mitigate global planning absence, GlobalRAG introduces planning-aware optimization that explicitly decomposes questions into structured subgoals, guided by a *Planning Quality Reward* measuring both structural and semantic consistency. To address unfaithful execution, we further design a *SubGoal Completion Reward* that encourages faithful alignment between planned subgoals and executed retrieval–reasoning steps. These objectives are jointly optimized using GRPO (Shao et al., 2024), enabling stable and coherent long-horizon reasoning. Additionally, inspired by TIRESRAG-R1 (He et al., 2025), we adopt a progressive weight annealing strategy to balance process-level supervision and final-answer optimization during training.

Experiments across diverse in-domain and out-of-domain benchmarks demonstrate that GlobalRAG consistently outperforms strong baselines, yielding substantial gains in multi-hop reasoning accuracy.

Our contributions can be summarized 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 requires multi-step reasoning over evidence from multiple sources (Li and Du, 2023; Zhang et al., 2024; Cheng et al., 2025). Retrieval-augmented generation (RAG) addresses this challenge by coupling retrieval with generation (Ji et al., 2023; Feng et al., 2024; Shi et al., 2024). Existing RAG methods for multi-hop QA mainly fall into two categories. Graph-structured approaches (Gutiérrez et al., 2025; Gutiérrez et al., 2025; Fang et al., 2025) organize textual knowledge into entity–relation graphs and aggregate evidence along multi-hop paths, but require explicit graph construction and maintenance (Gutiérrez et al., 2025; Gutiérrez et al., 2025). Iterative RAG approaches (Trivedi et al., 2023; Shao et al., 2023; Liu et al., 2024; Jiang et al., 2025; Li et al., 2025) perform retrieval and reasoning in multiple steps (e.g., IRCOT (Trivedi et al., 2023), Search-o1 (Li et al., 2025)), yet heavily rely on LLM-generated intermediate thoughts and are thus prone to hallucination (Liu et al., 2024; Jiang et al., 2025). In contrast, GlobalRAG leverages reinforcement learning to enhance global planning and coordinated reasoning, enabling more accurate evidence localization and integration for multi-hop QA.

**Large Language Models and Reinforcement Learning.** Reinforcement learning has substantially 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 recently been 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) adopts outcome-based rewards to encourage retriever usage, while StepSearch (Wang et al., 2025) scores per-step retrieval quality via information gain and redundancy penalties. However, lacking explicit global planning and suffering from unfaithful execution, these methods remain suboptimal for multi-hop QA (Jin et al., 2025a; Wang et al., 2025). By contrast, we introduce a Planning Quality Reward and a Subgoal Completion Reward to explicitly reinforce global reasoning and structured progress in multi-hop QA.

## 3 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

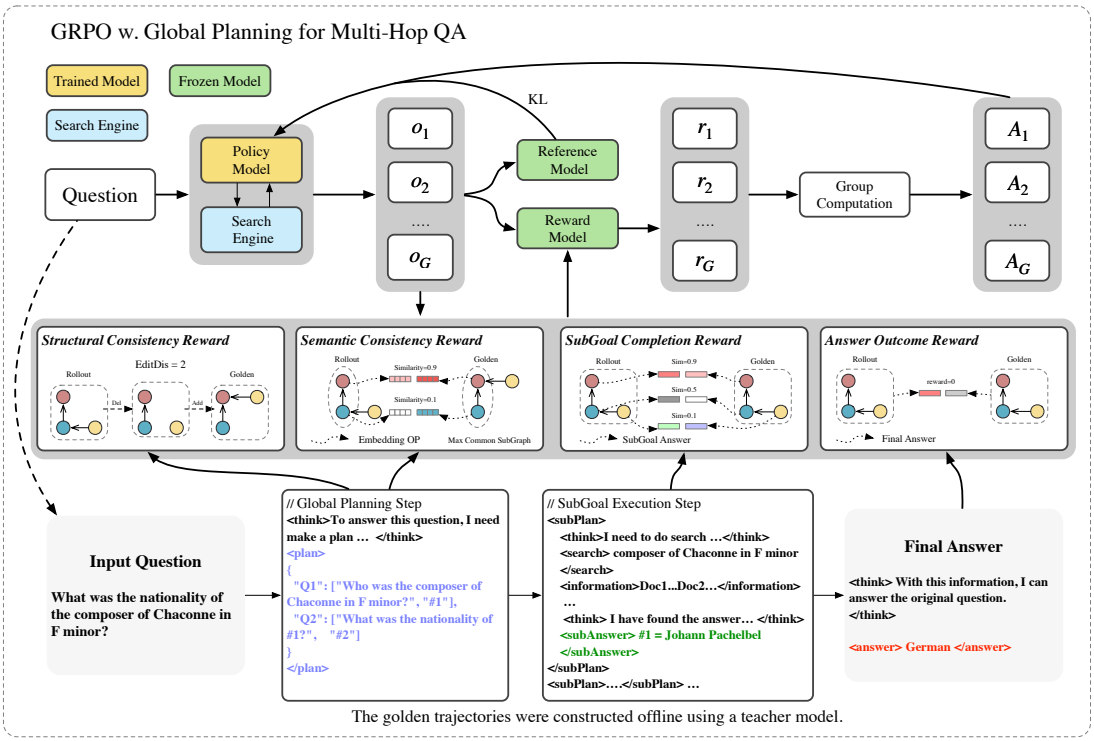



Figure 2: Overview of GlobalRAG. The top illustrates rollout retrieval–enhanced GRPO with a search engine; the bottom shows golden trajectories constructed by a teacher model; and the middle depicts the global planning–aware reward design used to update the policy model.

of the framework is illustrated in Figure 2, while detailed formulations and implementation specifics of the GRPO training procedure are deferred to Appendix C.

### 3.1 Golden Trajectory Generation with Global Planning

In this section, we describe the offline generation of golden trajectories, as illustrated in Figure 2(bottom). The generation process proceeds through three stages: *global planning*, *subgoal execution*, and *final answer generation*. For a detailed description of the trajectory generation procedure, including implementation specifics, see Appendix D.

### 3.2 Global Planning–Aware Reward Design

While reinforcement learning has been shown effective for RAG, existing approaches that rely primarily on simple rule-based rewards (such as final-answer or format rewards) tend to encourage shallow retrieval and short-horizon optimization, particularly in multi-hop QA settings where explicit global planning signals are absent. This often leads the model to deviate from the original reasoning intent during intermediate steps. To mitigate this issue, we propose *Global Planning–Aware Rewards*, which provide process-level supervision to better align long-horizon reasoning with the overall task objective. Rather than enforcing strict imitation of a single reference plan, our design adopts a soft, similarity-based alignment mechanism that assigns partial credit to diverse yet valid planning structures, **which we empirically observe to vary in subgoal ordering and granularity while remaining task-consistent**. This formulation promotes flexible but faithful reasoning and reduces overfitting to specific planning patterns. Concretely, our approach integrates three complementary components: *Structural Consistency*, *Semantic Consistency*, and *Subgoal Completion* rewards.

#### 3.2.1 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:

$$R_{\text{str}} = \exp\big(-\frac{d_{\text{edit}}(G, \tilde{G})}{N_{\tilde{G}} + E_{\tilde{G}}}\big), \tag{1}$$

where $d_{\text{edit}}(G, \tilde{G})$ denotes the graph edit distance. $N_{\tilde{G}}$ and $E_{\tilde{G}}$ denote the node and edge counts of $\tilde{G}$, respectively.

**Semantic Consistency Reward** evaluates the semantic alignment between the predicted plan and the golden plan. First, we employ a depth-first search (DFS) algorithm to identify the maximum common subgraph between the generated plan and the reference plan, thereby establishing valid and reliable structural correspondences. Second, we use the E5-base (Wang et al., 2022) model to embed all subproblem nodes into vectors, and compute semantic similarity only for matched subproblems within the maximum common 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:

$$R_{\text{sem}} = \frac{1}{|\mathcal{P}_{\text{gold}}|} \sum_{i=1}^{|\mathcal{M}_P|} \cos\Big(\rho^{(i)}_{\text{pred}}, \rho^{(i)}_{\text{gold}}\Big), \tag{2}$$

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

#### 3.2.2 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 (e.g., the F1 score between the predicted and gold answers). The reward is normalized by $|\mathcal{P}_{\text{gold}}|$ to ensure coverage, with unmatched nodes contributing zero:

$$R_{\text{step}} = \frac{1}{|\mathcal{P}_{\text{gold}}|} \sum_{i=1}^{|\mathcal{M}_P|} \phi\Big(o^{(i)}_{\text{pred}}, o^{(i)}_{\text{gold}}\Big). \tag{3}$$

#### 3.2.3 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:

$$R_{\text{answ}} = \begin{cases} 1, & \text{if the answer exactly matches,} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

#### 3.2.4 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:

$$R_{\text{form}} = \begin{cases} 1, & \text{if the format is fully compliant,} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

#### 3.2.5 Progressive weight annealing strategy

Inspired by prior work, we adopt a progressive annealing scheme of *structure first, accuracy later*.

| 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.

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:

$$w_t = \frac{1}{1 + \exp\left(\frac{t - 0.9T}{10}\right)}, \tag{6}$$

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

The overall reward integrates all components with dynamic weighting:

$$R(t) = w_t\big(\alpha R_{\text{form}} + \lambda R_{\text{str}} + \gamma R_{\text{sem}} + \delta R_{\text{step}}\big) + R_{\text{answ}}, \tag{7}$$

This curriculum smooths optimization, reduces variance, coordinates retrieval under the global plan, and transitions from structural learning to end-to-end reasoning accuracy.

## 4 Experiments

### 4.1 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. Detailed statistics and corpus information are provided in Appendix E.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. Details are provided in Appendix E.2.

**Training and Implementation Details.** We conduct experiments on Qwen-2.5(3B/7B) models

(Base/Instruct) and use a dense retriever retrieving three passages for all retrieval-based methods. Training data are merged from HotpotQA, 2WikiMultiHopQA, and MuSiQue, generating 8,394 supervised trajectories for GlobalRAG. See Appendix E.3 for more details.

### 4.2 Main Results

Table 1 compares GlobalRAG with competitive RL-trained and non-RL baselines across five datasets, with the following observations:

**(1) Overall SOTA with strong gains:** GlobalRAG consistently outperforms all baselines, achieving the highest F1 on all datasets and four best/one second-best EM scores. Compared with StepSearch, average F1 improves by 14.2% (HotpotQA +9.7%, 2Wiki +23.5%, MuSiQue +5.1%, Bamboogle +17.9%, WikiHop +5.6%), demonstrating the effectiveness of our planning-aware RL framework in multi-hop QA.

**(2) Gains correlate with dataset structure and multi-hop necessity:** Performance improvements vary with dataset design and multi-hop requirements. WikiHop and HotpotQA include many questions solvable with single-hop evidence, where extra retrieval can add noise (Chen and Durrett, 2019). In contrast, 2Wiki's cross-document dependencies make iterative planning more effective. MuSiQue requires complex multi-hop reasoning; GlobalRAG outperforms Search-R1, though gains over StepSearch are smaller since it already excels on its training distribution. Overall, the method shows the largest advantage when multi-hop reasoning and explicit cross-document aggregation are essential.

**(3) Robust in-domain and out-of-domain performance:** GlobalRAG leads on both in-domain (HotpotQA, 2Wiki, MuSiQue) and out-of-domain datasets (Bamboogle, WikiHop), indicating strong generalization. By contrast, Search-R1 performs well in-domain but degrades notably out-of-domain, highlighting generalization challenges of existing methods.

**(4) Efficiency in Data, Training, and Inference-Time Trade-offs:** Using only 8k examples (42% of StepSearch's 19k; 4.7% of Search-R1's 170k), GlobalRAG achieves both 14.2% EM/F1 gains over StepSearch and 28.2% EM, 29.4% F1 gains over Search-R1. This efficiency in data utilization and inference-time performance–cost trade-offs is mainly attributed to planning-aware supervision, including the Planning Quality and SubGoal Completion Rewards. As illustrated in Figure 3 (where bubble size indicates the efficiency ratio), GlobalRAG establishes a superior balance between reasoning performance and token cost on 2WikiMultihopQA, with detailed analyses for the remaining datasets deferred to Appendix F.

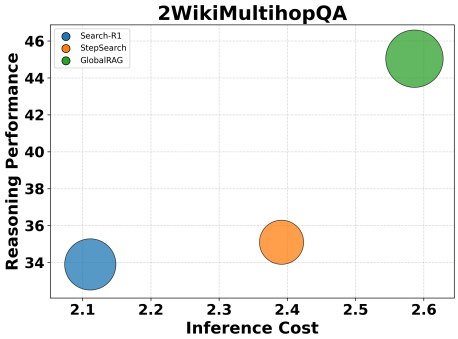



Figure 3: Inference-time performance versus token cost for three reinforcement learning based methods on **2WikiMultihopQA**.

**(5) Generalization across backbones and scales:** Across Qwen2.5 3B/7B Base and Instruct models, GlobalRAG consistently tops all baselines, and performance margins are preserved or widened with larger models. Moreover, as shown in Appendix L, consistent gains are also observed on the LLaMA-3.1-8B backbone, indicating that our method generalizes beyond the Qwen family to different model architectures and pretraining regimes.

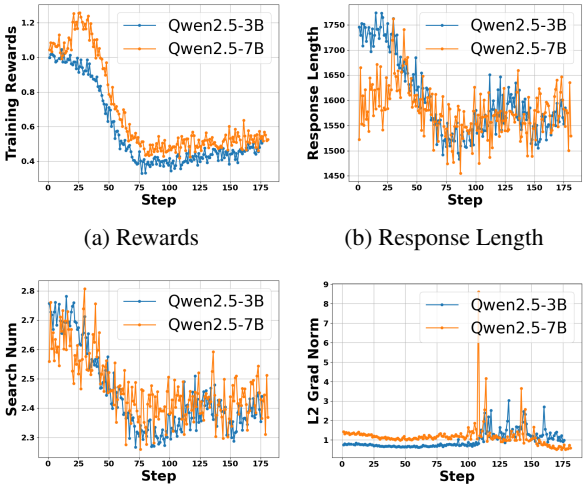
### 4.3 Model Scale Study

We use Qwen2.5 as the base model and conduct experiments at the 3B and 7B scales. As shown in Figure 4, the 7B model achieves higher reward values overall than the 3B model at the same training stage, and it also performs more searches with larger gradient norms. In the initial stage, the 7B model has shorter response lengths, but these gradually converge toward those of the 3B model during training. These trends indicate that, for larger-scale models, our method encourages the agent to retrieve information and optimize strategies more actively, while exhibiting more dynamic training behavior.

### 4.4 Ablation Study

We conduct a systematic ablation on GlobalRAG, and report the results in Table 2.

To validate global planning, we ablate both plan-level rewards jointly. *w/o* $R_{\text{str}}$ removes supervision over the global plan graph; *w/o* $R_{\text{sem}}$ removes su-




(a) Rewards (b) Response Length

(c) Search Num (d) Grad Norm

Figure 4: Comparison of Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct models across four metrics. From left to right and top to bottom: (a) Training Rewards; (b) Response Length; (c) Search Num; (d) L2 Gradient Norm.

| Methods | HotpotQA | | 2Wiki | | MuSiQue | | Bamboogle | | WikiHop | |
|---|---|---|---|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| w/o $R_{\text{sem}}$ | 30.0 | 41.5 | 38.1 | 44.1 | 10.1 | 17.2 | 37.6 | 51.4 | 10.4 | 18.1 |
| w/o $R_{\text{str}}$ | 30.6 | 42.2 | 39.6 | 46.0 | 9.5 | 18.3 | 39.2 | 47.9 | 11.2 | 18.7 |
| w/o $R_{\text{step}}$ | 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 2: Ablation study on reward components.

pervision over subgoal semantic composition. Both settings reduce EM/F1 across all datasets, with greater degradation for $R_{\text{sem}}$ than $R_{\text{str}}$. This confirms that plan-level guidance is essential beyond $R_{\text{step}}$: 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.

Removing intermediate subgoal supervision (*w/o* $R_{\text{step}}$) yields the largest performance declines (averaging 17.7% EM and 12.5% F1 drops). This proves that validating intermediate execution is essential to ensure faithful plan execution and prevent partial answers from becoming final predictions.

Disabling progressive reward-weight annealing (*w/o* $w_t$) also consistently degrades performance. This result suggests that the $w_t$-based curriculum stabilizes optimization by progressively shifting focus from early planning guidance to outcome-driven rewards, mitigating noise from imperfect

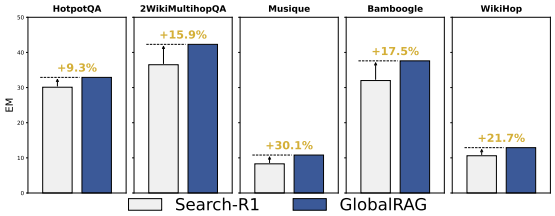



Figure 5: Comparison of the performance of our method GlobalRAG with the baseline Search-R1 on five benchmark datasets under identical datasets and unified experimental settings. (LLM: Qwen2.5-3B-Instruct)

teacher trajectories (see Appendix G for plan diversity analysis).

Finally, we conduct ablation studies on retrieval size (top-$k$) and GRPO group size, with results in Appendix H and I, showing optimal performance at $k = 3$ and group size 5. Sensitivity analyses of the reward weight $\lambda$, discount factor $\gamma$, and transition point $T$ (Appendix J and K) indicate optimal values of $\lambda = 0.5$, $\gamma = 0.5$, and $T = 50$. These findings highlight GlobalRAG's robustness to hyperparameter choices.

## 4.5 Algorithmic Attribution Study

| Methods | HotpotQA | | 2Wiki | | MuSiQue | | Bamboogle | | WikiHop | |
|---|---|---|---|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| **Qwen2.5-3b-Instruct** | | | | | | | | | | |
| Search-R1 | 30.1 | 36.1 | 36.5 | 40.8 | 8.3 | 12.7 | 32.0 | 33.5 | 10.6 | 13.3 |
| GlobalRAG | 32.9 | 44.2 | 42.3 | 47.8 | 10.8 | 18.6 | 37.6 | 49.3 | 12.9 | 20.7 |
| **Qwen2.5-7b-Instruct** | | | | | | | | | | |
| Search-R1 | 34.4 | 45.6 | 42.4 | 47.6 | 9.0 | 16.7 | 41.1 | 49.3 | 15.1 | 21.9 |
| GlobalRAG | 38.2 | 50.7 | 47.8 | 53.2 | 14.9 | 21.2 | 48.8 | 61.6 | 19.0 | 28.0 |
| **Llama3.1-8b-Instruct** | | | | | | | | | | |
| Search-R1 | 35.7 | 46.9 | 44.6 | 50.1 | 12.9 | 20.0 | 44.8 | 56.5 | 16.5 | 22.0 |
| GlobalRAG | 36.9 | 48.6 | 46.6 | 51.9 | 17.0 | 25.7 | 49.6 | 61.9 | 17.8 | 25.1 |

Table 3: Algorithmic attribution (controlled data): GlobalRAG vs. Search-R1.

To evaluate the intrinsic effectiveness of our proposed framework, we conducted an algorithmic attribution study comparing GlobalRAG against the Search-R1 baseline under strictly controlled experimental settings. As illustrated in Figure 5, GlobalRAG achieves a significant performance leap, specifically yielding a 9.3% relative improvement in EM score when utilizing the Qwen2.5-3B-Instruct backbone. This performance gain, achieved using identical training datasets and the GRPO reinforcement learning algorithm, underscores that the advancements are primarily driven by our architectural innovations in global reasoning and evidence aggregation rather than data scaling.

Notably, similar and consistent improve-

ments are also observed when switching to the LLaMA3.1-8B-Instruct backbone, where GlobalRAG consistently outperforms Search-R1 across all five benchmarks, indicating that the proposed framework generalizes well across different model families. Specifically, GlobalRAG demonstrates superior proficiency in handling complex multi-hop queries where structured knowledge integration is paramount. Detailed analyses are provided in Appendix L.

### 4.6 Different RL Methods: PPO vs. GRPO

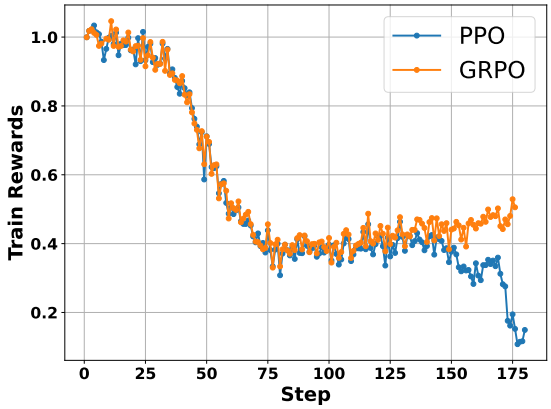



Figure 6: Training reward curves for PPO and GRPO. GRPO achieves higher rewards with faster convergence and more stable training.

| Methods | HotpotQA | | 2Wiki | | MuSiQue | | Bamboogle | | WikiHop | |
|---|---|---|---|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| PPO | 29.5 | 40.6 | 37.5 | 43.6 | 9.1 | 15.1 | 32.0 | 45.2 | 10.3 | 17.9 |
| GRPO | 32.9 | 44.2 | 42.3 | 47.8 | 10.8 | 18.6 | 37.6 | 49.3 | 12.9 | 20.7 |

Table 4: Performance comparison of PPO and GRPO within the GlobalRAG framework on five multi-hop QA benchmarks, measured by EM and F1 scores.

We compare PPO and GRPO for training GlobalRAG. As shown in Figure 6, GRPO attains higher training rewards and converges more steadily than PPO, indicating a more effective optimization process.

Table 4 reports the downstream QA results. GRPO consistently outperforms PPO across all benchmarks on both EM and F1, with more pronounced gains on HotpotQA, 2WikiMultihopQA, and Bamboogle, which require multi-hop evidence aggregation and complex reasoning. Improvements on MuSiQue and WikiHop, though smaller in magnitude, remain consistent. These results indicate that GRPO exhibits more stable training dynamics and superior generalization compared to PPO.

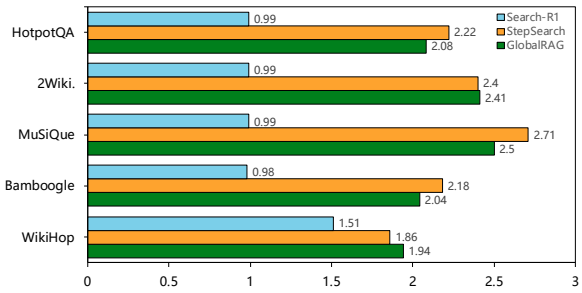



Figure 7: Comparison of average retrieval counts across datasets for Search-R1, StepSearch and GlobalRAG.

### 4.7 Retrieval Performance for Multi-hop QA

To explain why GlobalRAG achieves stronger multi-hop QA performance, we analyze its retrieval behavior. As shown in Figure 7, both GlobalRAG and StepSearch perform roughly twice as many retrievals as Search-R1 (about $1.2\times$ on WikiHop), indicating that incorporating process-oriented rewards, rather than relying solely on outcome-based rewards, more effectively encourages iterative retrieval and supports correct answer identification.

Notably, GlobalRAG uses a comparable but slightly smaller number of retrievals than StepSearch (Figure 7), while achieving higher EM and F1 scores (Table 1). This suggests that the planning quality and subgoal completion rewards improve global planning and faithful execution, leading to higher accuracy without increasing retrieval steps.

## 5 Conclusions

In this work, we presented GlobalRAG, a reinforcement learning framework for multi-hop question answering that explicitly enforces global planning and faithful execution. By modeling question decomposition as a structured plan and optimizing planning-quality and subgoal-completion rewards under a progressive annealing schedule, GlobalRAG aligns high-level reasoning intent with retrieval and execution behaviors. Extensive experiments across in-domain and out-of-domain benchmarks show that this plan-centric optimization consistently improves multi-hop reasoning accuracy without increasing retrieval cost. These results highlight the importance of supervising how reasoning unfolds, rather than optimizing final answers alone, and point to future work on more flexible planning and reasoning trajectories.

## Limitations

Our study has four main limitations. First, due to computational and cost constraints, we do not conduct reinforcement learning training on very large-scale models (e.g., 14B or 32B parameters). Whether the proposed framework can be effectively scaled to such model sizes while maintaining stable training dynamics and planning performance remains an open question. Second, our experiments focus on multi-hop question answering. We do not include a detailed analysis or dedicated training data generation for single-hop tasks, which may limit the applicability of our findings to other reasoning settings. Third, for complex multi-hop queries, multiple valid decomposition paths may exist. Although the similarity-based reward design allows partial credit for semantically aligned plans, the current training setup still optimizes toward a single reference trajectory. Explicitly modeling and optimizing over multiple valid planning paths is left for future work. Finally, the framework improves planning accuracy primarily through reinforcement learning during training, but does not incorporate an explicit mechanism for real-time error correction at inference. If the model produces an incorrect initial decomposition, it cannot revise the global plan based on intermediate retrieval failures or reasoning inconsistencies, which remains a challenge for achieving more robust and adaptive multi-hop reasoning.

## References


Jifan Chen and Greg Durrett. 2019. Understanding dataset design choices for multi-hop reasoning. *arXiv preprint arXiv:1904.12106*.

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.

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*.

Qichuan Liu, Chentao Zhang, Chenfeng Zheng, Guosheng Hu, Xiaodong Li, and Zhihong Zhang. 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 (Volume 1: Long Papers)*, pages 23433–23456.

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.

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.

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 stepwise 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.

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*.

# A Prompt for Rollout

The prompt we use for trajectory generation is shown in Figure 8.

**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 8: The prompt used for trajectory generation

# B Analysis of RL-based RAG Problems

Following this analysis, we present a quantitative breakdown of Search-R1's failure modes across datasets. From up to 300 sampled instances per dataset (1325 instances in total, including all 125 instances from Bamboogle), Search-R1 fails on 1072 cases. Among these errors, 44.96% are attributed to the absence of global planning, while 28.45% result from unfaithful execution.

Specifically, we sampled 300 instances from each of five public multi-hop reasoning benchmarks—HotpotQA, 2WikiMultiHopQA (2Wiki), MuSiQue, Bamboogle, and WikiHop. For Bamboogle, which consists of only 125 instances, we evaluated on the full set. 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 Detailed GRPO Adaptation for GlobalRAG

## C.1 Token-level Policy Optimization with Masked GRPO

To integrate GRPO with our GlobalRAG, we adopt the token-masking strategy from Search-R1 (Jin et al., 2025a), where all tokens between `<information>` and `</information>` are masked. This ensures that the policy gradient is computed

solely on tokens generated by the language model, excluding retrieved or externally provided content from optimization.

The GRPO training objective is defined as follows:

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\Bigg[ \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Bigg(\frac{\pi_\theta(o_{i,t}\mid q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid q,o_{i,<t})}\hat{A}_{i,t}, \text{clip}\Big(\frac{\pi_\theta(o_{i,t}\mid q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid q,o_{i,<t})}, 1-\varepsilon, 1+\varepsilon\Big)\hat{A}_{i,t}\Bigg) - \beta D_{\text{KL}}\big[\pi_\theta \parallel \pi_{\text{ref}}\big]\Bigg] \quad (8)$$

Here, $\hat{A}_{i,t}$ denotes the normalized token-level advantage for the $t$-th token in the $i$-th trajectory within a group of size $G$, computed as

$$\hat{A}_{i,t} = \frac{R_{i,t} - \text{mean}(R_t)}{\text{std}(R_t)}.$$

The parameter $\varepsilon$ controls the clipping range for the policy ratio, and $\beta$ weights the KL divergence regularization term that constrains the updated policy $\pi_\theta$ toward the reference policy $\pi_{\text{ref}}$.

This formulation enables GRPO to optimize the actor model over multiple rollouts while stabilizing training and ensuring that only LLM-generated tokens contribute to the policy gradient.

### C.2 Single-model ReAct Instantiation

Our method is instantiated under a ReAct-style single-model framework, where a single language model parameterized by $\theta$ jointly performs reasoning, query generation, and answer synthesis. All intermediate reasoning steps, search queries, and final answers are generated autoregressively by the same policy $\pi_\theta$, conditioned on the evolving interaction history.

This unified formulation allows GRPO to assign token-level credit consistently across reasoning and action tokens within a shared policy space, avoiding credit fragmentation across multiple modules.

### C.3 RL with Search Engine for Multi-hop Question Answering

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 final answer:

$$p(a \mid q,\mathcal{C}) \approx \sum_{r_{1:K},\,D_{1:K}} \Bigg\{ \prod_{i=1}^{K}\Big[\pi_\theta\big(r_i \mid q, D_{<i}\big) \cdot p_{\text{se}}\big(D_i \mid r_i,\mathcal{C}\big)\Big]\Bigg\} p_\phi\big(a \mid q, D_{1:K}\big). \quad (9)$$

**Training objective.**

$$\max_{\theta,\phi}\ \mathbb{E}_{(q,a^\star)\sim\mathcal{D}}\,\mathbb{E}_{\tau\sim\pi_\theta(\cdot|q,\mathcal{C})}\big[\,R(q,\tau,a^\star;\phi)\,\big] - \beta\,\mathrm{D}_{\text{KL}}\Big[\pi_\theta(\cdot \mid q,\mathcal{C}) \,\Big\|\, \pi_{\text{ref}}(\cdot \mid q,\mathcal{C})\Big]. \quad (10)$$

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 optionally applied to stabilize policy updates.

## D Golden Trajectory Generation with Global Planning

To provide high-quality supervision for learning planning-aware reasoning policies, we construct *golden reasoning trajectories* that explicitly encode both global task decomposition and step-by-step execution. These trajectories serve as expert demonstrations for evaluating and optimizing model rollouts, ensuring alignment between high-level planning decisions and local reasoning actions. Throughout the entire trajectory construction process, GPT-4o is used as the underlying model.

The trajectory construction process is performed offline and consists of three main stages.

### D.1 Global Planning

Given an input multi-hop question, a fixed teacher model, operating in an instruction-following mode with frozen parameters, first performs global task analysis to derive an explicit execution plan.

Specifically, the teacher model conducts a semantic analysis of the input query to identify its logical structure, constraints, and underlying reasoning type (e.g., bridge, comparison, or compositional reasoning). Based on this analysis, the original question is systematically decomposed into a sequence of fine-grained subqueries, each formulated as a natural-language question that can be independently resolved via retrieval and reasoning.

Each subquery is associated with a placeholder identifier that represents its expected intermediate result. These placeholders are subsequently referenced by downstream subqueries to explicitly capture information dependencies. The resulting subqueries and their dependency relations are organized into a structured representation, such as a task dependency graph, which defines a partial order over subgoals and serves as a global execution plan.

The entire plan is generated in a single forward pass of the teacher model using a predefined planning prompt, and remains fixed throughout subsequent stages, ensuring consistency and reproducibility.

### D.2 Subgoal Execution

Given the global plan, the teacher model resolves each subtask in topological order according to the dependency structure.

For each subquery, the model performs localized reasoning, including evidence retrieval, information synthesis, and intermediate answer generation. Retrieval is conducted using the same dense retriever employed during training and inference, without access to oracle documents, to ensure that the generated trajectories reflect realistic reasoning conditions.

Once a subtask is completed, its generated answer is substituted into the corresponding placeholder and propagated to all dependent subqueries. This explicit result propagation mechanism ensures that later reasoning steps operate on an updated context that incorporates previously resolved information, thereby maintaining global coherence.

Throughout this process, retrieval queries are conditioned on both the original question and the current intermediate context, enabling targeted evidence acquisition for each subgoal. The execution of each subtask is recorded as a sequence of state–action–result tuples, forming a fine-grained trace of the reasoning process.

### D.3 Final Answer Generation

After all subtasks have been resolved, the intermediate answers are integrated according to the dependency graph to produce a globally consistent final answer.

Additionally, to ensure the reliability and internal consistency of the constructed golden trajectories, we apply a set of strict filtering criteria throughout the generation process. First, at the global planning stage, if the execution plan produced by the teacher model violates the predefined topological constraints of the task dependency graph—e.g., introducing cyclic dependencies or inconsistent subgoal ordering—the corresponding sample is discarded. Second, during subgoal execution, each subquery is allowed up to 15 retrieval attempts; if the teacher model fails to derive a valid intermediate answer within this budget, the entire trajectory is removed to avoid propagating incomplete or unstable reasoning traces. Finally, after final answer generation, trajectories whose predicted answers do not exactly match the ground-truth annotations are excluded. Together, these filtering rules prioritize structural correctness, execution feasibility, and answer-level accuracy, resulting in a high-quality set of expert demonstrations suitable for training and evaluation. Detailed prompt we use can be found in PER-PSE (Liu et al., 2025).

## E Experimental Details

### E.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 work (Jin et al., 2025b), and deploy the system as an online retrieval service using FastAPI for efficient query processing.

The training set for GlobalRAG is built from the training splits of HotpotQA, 2WikiMultiHopQA, and MuSiQue, totaling 8,394 examples, with golden plans and golden graphs precomputed for all training data. HotpotQA, 2WikiMultiHopQA, MuSiQue, and Bamboogle are sourced from the FlashRAG Datasets collection. To prevent data leakage, we deduplicate instances where the test or development sets overlap with our training data. 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, 125 from Bamboogle, and 5,129 from WikiHop. We report results on the full test sets for Bamboogle and WikiHop, and on the sampled subsets for HotpotQA, 2WikiMultiHopQA, and MuSiQue. For datasets with non-public test sets (e.g., HotpotQA, 2WikiMultiHopQA, MuSiQue, WikiHop), we use their development sets as proxies

for evaluation and report the corresponding results.

### E.2 Metrics

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

#### E.2.1 Exact Match (EM)

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

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

#### E.2.2 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.

### E.3 Training and retrieval Details

Experiments use Qwen-2.5-3/7B (Base/Instruct) models. For retrieval, we build a `faiss-gpu` index over the 2018 Wikipedia snapshot and use E5 embeddings as the dense retriever. Following Search-R1, we retrieve 3 passages for all methods. Training merges HotpotQA, 2WikiMultiHopQA, and MuSiQue. Global-plan supervision for GlobalRAG is derived via subgoal–dependency plans, verifying intermediate steps, yielding 8,394 supervised trajectories.

### E.4 Training Hyperparameters

We trained our model on 4× 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 $1 \times 10^{-6}$, 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.

For the progressive reward scheduling described in the main text, we adopt a curriculum that gradually shifts emphasis from process-level supervision to final answer accuracy. The scheduling weight is defined as

$$w_t = \frac{1}{1 + \exp\left(\frac{t - 0.9T}{10}\right)}, \tag{15}$$

where $t$ denotes the current training step, and $T$ is a hyperparameter that determines the transition point, set to $T = 50$. The process reward aggregates format, structural, semantic, and step-level components with coefficients $\alpha = 0.1$ and $\lambda = \gamma = \delta = 0.5$, respectively, while the answer reward is applied without additional scaling. These hyperparameters are fixed across all experiments unless otherwise specified.

## F Performance–Reasoning Cost Trade-off Study

### F.1 Inference-Time Performance–Cost Trade-off

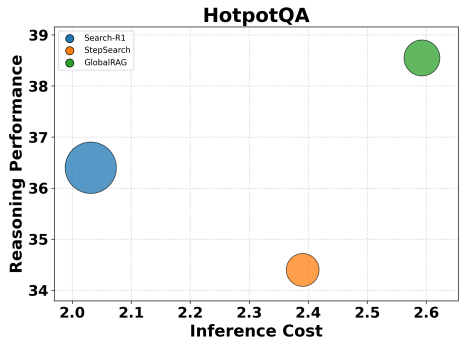



Figure 9: Inference-time performance versus token cost for three reinforcement learning based methods on **HotpotQA**.

To systematically evaluate the efficiency and scalability of **GlobalRAG**, we conduct a comparative study against two competitive baselines: **Search-R1** and **StepSearch**. All methods are implemented using the QWEN2.5-3B-INSTRUCT model as the backbone, ensuring a fair assessment of reasoning strategies independent of model scale. We evaluate the methods across five representative multi-hop reasoning datasets: HOTPOTQA, BAMBOOGLE, MUSIQUE, 2WIKIMULTIHOPQA, and WIKIHOP. Reasoning performance (y-axis) is measured as the average of *Exact Match (EM)* and

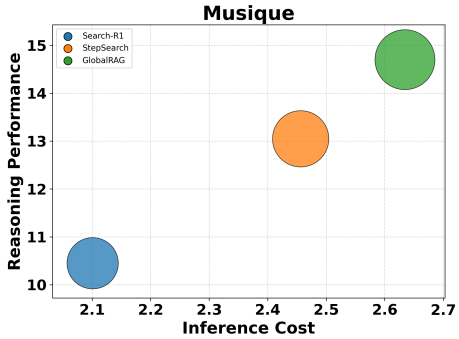



Figure 10: Inference-time performance versus token cost for three reinforcement learning based methods on **Musique**.

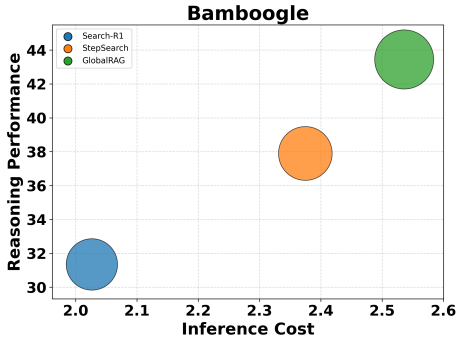



Figure 11: Inference-time performance versus token cost for three reinforcement learning based methods on **Bamboogle**.

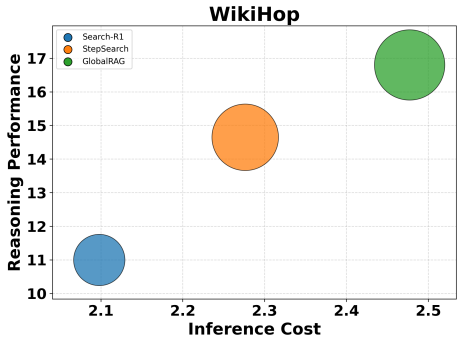



Figure 12: Inference-time performance versus token cost for three reinforcement learning based methods on **WikiHop**.

*F1* scores. Inference cost (x-axis) is represented by the logarithm of total token usage per query, $\log_{10}(\text{Tokens})$. Bubble size encodes the efficiency ratio, defined as:

$$\frac{\text{Average(EM, F1)}}{\log_{10}(\text{Tokens})} \tag{16}$$

where larger bubbles indicate higher performance per unit of token expenditure.

The figures here, alongside Figure 3 in the main text, illustrate the trade-off between reasoning accuracy and computational cost across all datasets. Several key observations emerge. First, **GlobalRAG** (green bubbles) consistently occupies the top-right quadrant, demonstrating the highest reasoning performance. For instance, as shown in Figure 3 for 2WIKIMULTIHOPQA, GlobalRAG achieves a score above 45, whereas the baselines remain below 36, highlighting the effectiveness of global evidence integration and reasoning path optimization for complex multi-hop queries. Second, although GlobalRAG requires more tokens, its efficiency ratio remains comparable to or exceeds that of the baselines in most datasets, notably BAMBOOGLE and WIKIHOP, suggesting that performance gains arise from high-quality reasoning rather than brute-force token usage. Finally, the baselines show distinct limitations: **Search-R1** (blue) maintains the lowest inference cost but suffers from a substantial performance gap, often plateauing due to limited capability in capturing inter-document dependencies. **StepSearch** (orange) lies between Search-R1 and GlobalRAG in cost, yet its performance improvements are not always proportional—for example, on HOTPOTQA, it requires more tokens but performs slightly worse than Search-R1.

Overall, this trade-off study demonstrates that **GlobalRAG** establishes a new Pareto frontier among retrieval-augmented generation systems. By strategically leveraging additional tokens for global context integration, it overcomes performance limitations inherent in step-wise or iterative search methods, offering the most robust scaling behavior for 3B-parameter models when reasoning accuracy is paramount.

### F.2 Training Efficiency under Reinforcement Learning

Table 5: Training Efficiency Comparison under Reinforcement Learning Framework

| Method | RL-based | Batch Size | Training Steps | Training Data Size |
|---|---|---|---|---|
| Search-R1 | ✓ | 512 | 500 | 170K |
| StepSearch | ✓ | 256 | 500 | 19K |
| GlobalRAG | ✓ | 256 | **200** | **8K** |

Beyond inference-time efficiency, we further examine the training efficiency of the three methods under a unified reinforcement learning framework, as summarized in Table 5. To ensure a fair comparison, all methods adopt the same batch size of 256. **Search-R1** is trained for 500 optimization steps

with approximately 170K training samples, while **StepSearch** follows the same number of training steps but relies on a smaller dataset of 19K samples. In contrast, **GlobalRAG** converges substantially faster, requiring only 200 training steps and merely 8K training samples.

This comparison reveals a clear efficiency advantage of **GlobalRAG** during training. Despite significantly lower data and optimization budgets, GlobalRAG achieves superior or comparable reasoning performance, indicating more effective reward utilization and policy learning. When considered jointly with the inference-time trade-off illustrated in Figures 9–12, these results demonstrate that GlobalRAG consistently establishes a more favorable efficiency frontier than Search-R1 and StepSearch across both training and inference stages.

### F.3 Reward Computation Overhead Analysis

Table 6: Per-step Wall-clock Time Breakdown under Different Reward Designs (Qwen2.5-7B-Instruct)

| Component | Search-R1 Reward | | GlobalRAG Reward | |
|---|---|---|---|---|
| | Time (s) | Ratio (%) | Time (s) | Ratio (%) |
| Rollout Generation | 186.82 | 50.69 | 289.89 | 42.30 |
| Reward Computation | 1.07 | 0.29 | 112.12 | 16.36 |
| Policy Update | 94.28 | 25.58 | 145.24 | 21.19 |
| Total per RL Step | 368.55 | 100.0 | 685.40 | 100.0 |

Table 7: Per-token Normalized Computation Cost per RL Step

| Reward Design | Rollout Time / Token (ms) | Reward Time / Token (ms) | Update Time / Token (ms) |
|---|---|---|---|
| Search-R1 Reward | 23.61 | 0.12 | 10.54 |
| GlobalRAG Reward | 28.67 | 7.00 | 9.06 |

To better contextualize the training efficiency gains reported in the previous subsection, we further analyze the wall-clock overhead introduced by different reward designs. Tables 6 and 7 report a fine-grained runtime decomposition under controlled experimental settings: **Search-R1** and **GlobalRAG** share the same QWEN2.5-7B-INSTRUCT backbone, identical reinforcement learning pipeline, and the same GlobalRAG-based training dataset. The sole difference lies in the reward formulation.

As shown in Table 6, rollout generation dominates per-step runtime for both reward designs, accounting for over 40% of total wall-clock time. Although GlobalRAG employs a more structured and computation-intensive reward, reward evaluation remains a secondary contributor compared to rollout sampling and policy optimization. Specifically, reward computation under GlobalRAG constitutes 16.36% of the per-step cost, while rollout generation and policy updates together still account for more than 60% of total runtime.

The per-token normalized analysis in Table 7 further clarifies this behavior. GlobalRAG incurs a higher reward computation cost per token (7.00 ms), which can be attributed to graph-aware matching and semantic similarity assessment. However, this overhead is amortized by the substantially larger cost of rollout generation. Notably, rollout and policy update costs per token remain comparable across reward designs, indicating that planning-aware rewards do not alter the fundamental computational profile of reinforcement learning optimization.

Taken together, these results demonstrate that the superior training efficiency of GlobalRAG, as shown in Table 5, does not stem from a cheaper reward signal. Instead, GlobalRAG achieves faster convergence and lower data requirements despite a moderately higher reward computation cost. This confirms that its efficiency gains arise from more informative and effective supervision, rather than reduced computational complexity, reinforcing the practical scalability of GlobalRAG under standard reinforcement learning regimes.

## G Quantitative Analysis of Plan Decomposition Diversity

To further examine whether GlobalRAG relies on a single predefined reasoning trajectory or is able to generate diverse plan decompositions, we conduct a quantitative analysis on the diversity of planning behaviors at inference time. This study is performed on the BAMBOOGLE dataset using a QWEN2.5-3B-INSTRUCT model fine-tuned under the GlobalRAG framework.

For each question, we sample five independent responses with identical decoding configurations. In total, this results in 625 generated answers for 125 questions. Among these generations, 254 answers are judged to be correct according to the dataset evaluation protocol. To assess plan-level diversity, we further group correct answers by their corresponding plan decompositions, where a plan is defined as the sequence of intermediate reasoning steps and sub-question structures produced by the model. Duplicate plans for the same question are collapsed into a single instance.

After deduplication, we obtain 144 distinct correct plan decompositions. This indicates that a substantial portion of correct answers arises from multiple, non-identical reasoning plans rather than repeated execution of a single trajectory. Quantitatively, over 56% of correct generations correspond to distinct plan decompositions, suggesting that the learned policy is capable of exploring alternative reasoning structures that still lead to valid answers.

These results provide empirical evidence that GlobalRAG is not strictly constrained to reproducing a single golden plan during inference. Instead, the model demonstrates a degree of flexibility in plan construction, generating multiple viable decompositions for the same question. While repeated plans remain common—reflecting stable high-reward reasoning patterns—the presence of a large number of distinct correct plans indicates that the planning policy learned by GlobalRAG is not limited to a fixed, deterministic trajectory.

We emphasize that this analysis is intended to establish the existence of plan-level diversity rather than to claim optimal or exhaustive exploration of the plan space. Nonetheless, the observed diversity supports the view that GlobalRAG can move beyond rigid adherence to predefined reasoning paths and instead learn a more flexible planning behavior under reinforcement learning.

## H Number of Retrieved Passages Study

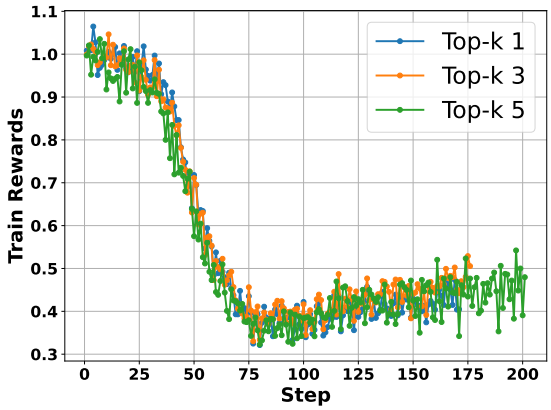



Figure 13: Comparison of training rewards under different numbers of retrieved passages ($k$). The figure reports the evolution of training rewards over steps when varying the top-$k$ retrieved passages ($k \in \{1, 3, 5\}$). (LLM: Qwen2.5-3B-instruct, RL: GRPO)

We conduct a systematic analysis of GlobalRAG under different retrieval depths $k$ on both In-Domain and Out-of-Domain multi-hop question answering benchmarks. The results are summarized in Table 8. Overall, model performance exhibits a consistent upward trend as the retrieval depth increases.

| Methods | HotpotQA | | 2Wiki | | Musique | | Bamboogle | | WikiHop | |
|---|---|---|---|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| $k = 1$ | 28.9 | 40.1 | 38.4 | 43.7 | 8.0 | 14.2 | 30.4 | 37.7 | 12.6 | 20.6 |
| $k = 3$ | 32.9 | 44.2 | 42.3 | 47.8 | 10.8 | 18.6 | 37.6 | 49.3 | 12.9 | 20.7 |
| $k = 5$ | 32.4 | 44.0 | 43.2 | 48.6 | 9.9 | 16.9 | 39.2 | 52.8 | 13.4 | 20.9 |

Table 8: Performance of Qwen2.5-3B-Instruct under different retrieval depths ($k$).

On typical In-Domain multi-hop reasoning datasets such as HotpotQA and 2WikiMultihopQA, increasing the retrieval depth from $k = 1$ to $k = 3$ leads to substantial improvements in both EM and F1. For instance, on HotpotQA, EM improves from 28.9 to 32.9, while F1 increases from 40.1 to 44.2. When the retrieval depth is further expanded to $k = 5$, performance remains at a relatively high level but the gains become marginal, suggesting that a moderate retrieval depth is sufficient to capture most of the relevant evidence required for multi-hop reasoning.

In contrast, on the more challenging Musique dataset, increasing $k$ results in only limited performance gains. This behavior may be attributed to the dataset's higher intrinsic difficulty or to a more dispersed distribution of supporting evidence, where simply retrieving additional passages does not substantially alleviate the reasoning burden.

For Out-of-Domain evaluation, the model is generally less sensitive to retrieval depth than in In-Domain settings, yet still demonstrates consistent improvements as $k$ increases. On datasets such as Bamboogle and WikiHop, both EM and F1 improve steadily when $k$ is increased from 1 to 3 and further to 5, indicating that expanding the retrieval context helps compensate for domain shifts by providing additional external knowledge. Notably, on Bamboogle, F1 rises from 37.7 to 52.8, highlighting that deeper retrieval can partially mitigate information sparsity caused by distributional mismatch.

It is also worth noting that the performance gains exhibit a clear diminishing-returns pattern. Across both In-Domain and Out-of-Domain benchmarks, a retrieval depth of $k = 3$ is often sufficient to cover the majority of useful evidence, while further increasing $k$ yields relatively limited benefits. This phenomenon may be attributed to the global planning mechanism employed by GlobalRAG: guided

by reinforcement learning signals, the model learns to organize subgoals and utilize retrieved evidence more effectively under a smaller retrieval budget, thereby reducing reliance on excessive or redundant documents.

Overall, these results demonstrate that GlobalRAG achieves stable and consistent improvements across different retrieval depths. The study further confirms that, for multi-hop question answering, appropriately increasing the retrieval depth can enhance reasoning performance, while the reinforcement learning–driven global planning mechanism improves retrieval efficiency and achieves a favorable balance between performance and retrieval cost.

## I Group Size Study in GRPO

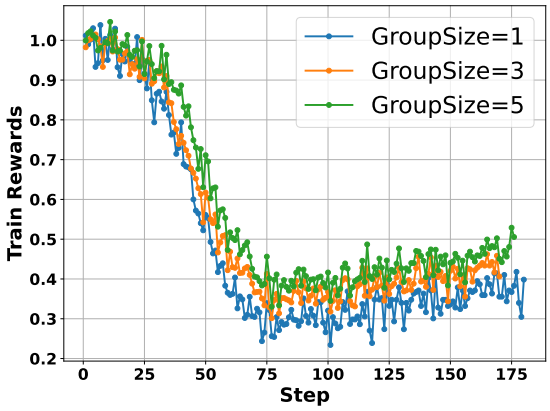



Figure 14: Comparison of GRPO training dynamics under different group sizes. (LLM: Qwen2.5-3B-instruct, RL: GRPO)

| Methods | HotpotQA | | 2Wiki | | Musique | | Bamboogle | | WikiHop | |
|---|---|---|---|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| $n = 1$ | 27.2 | 38.0 | 37.3 | 42.4 | 8.9 | 16.5 | 31.2 | 42.4 | 8.5 | 16.1 |
| $n = 3$ | 30.3 | 42.1 | 37.3 | 43.4 | 9.3 | 16.5 | 32.8 | 45.6 | 10.7 | 18.4 |
| $n = 5$ | 32.9 | 44.2 | 42.3 | 47.8 | 10.8 | 18.6 | 37.6 | 49.3 | 12.9 | 20.7 |

Table 9: Performance of Qwen2.5-3B-Instruct with GRPO under different group sizes ($n$).

Table 9 reports the performance of GlobalRAG trained with GRPO under different group size (n) settings across multiple multi-hop question answering benchmarks. As n increases from 1 to 5, we observe consistent improvements in both EM and F1 on all datasets.

From an algorithmic perspective, this trend can be attributed to the statistical properties of advantage estimation in GRPO. Unlike policy optimization methods that rely on an explicit value function, GRPO estimates advantages using relative rewards within a sampled group, where the group-wise average reward serves as an implicit baseline. When the group size is small, this baseline estimate is more susceptible to noise from individual samples, resulting in higher variance in the advantage signal and less stable policy updates. Increasing the group size yields a more reliable estimate of the expected reward under the current policy, thereby providing a more accurate reference point for computing relative advantages.

This effect is particularly pronounced on reasoning-intensive benchmarks such as HotpotQA, 2Wiki, and WikiHop. For instance, on HotpotQA, increasing n from 1 to 5 leads to gains of 5.7 EM and 6.2 F1 points, indicating that a more stable relative advantage signal helps the model better discriminate among retrieval–reasoning trajectories of varying quality. On smaller or noisier datasets such as Musique and Bamboogle, the improvements are more modest but remain consistent, suggesting that larger group sizes primarily contribute to variance reduction during training rather than exploiting dataset-specific artifacts.

At the same time, the gains exhibit diminishing returns as the group size increases, highlighting a trade-off between the accuracy of baseline estimation and computational cost. Based on this observation, we adopt n=5 as the default group size in the remaining experiments unless otherwise specified.

## J Reward Weight Sensitivity Study

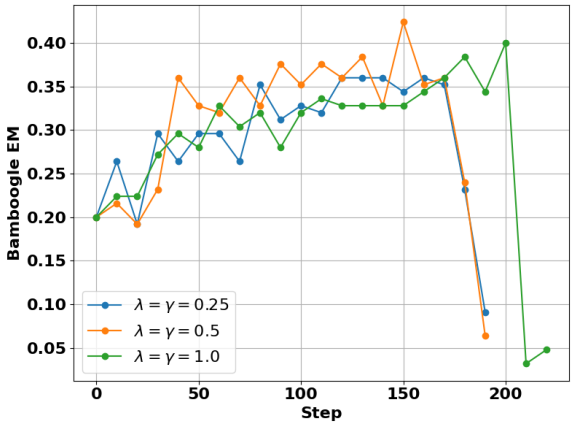



Figure 15: Validation EM on the Bamboogle dataset under different global-plan reward weight coefficients ($\lambda = \gamma$), evaluated every ten training steps. The plan reward jointly incorporates structural and semantic constraints.

| Weight | HotpotQA | | 2Wiki | | Musique | | Bamboogle | | WikiHop | |
|---|---|---|---|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| $\lambda = \gamma = 0.25$ | 31.4 | 42.2 | 40.9 | 46.9 | 10.6 | 18.2 | 36.0 | 48.7 | 11.7 | 19.3 |
| $\lambda = \gamma = 0.5$ | **32.9** | **44.2** | **42.3** | **47.8** | 10.8 | **18.6** | **37.6** | 49.3 | 12.9 | 20.7 |
| $\lambda = \gamma = 1.0$ | 32.2 | 43.8 | 41.4 | 47.2 | **11.1** | 18.0 | **37.6** | **49.6** | **13.1** | **21.5** |

Table 10: Performance comparison under different global-plan reward weight coefficients across five QA benchmarks, evaluated using Qwen2.5-3B-Instruct.

We investigate the effect of the global-plan reward weight by varying the coefficient $\lambda = \gamma \in \{0.25, 0.5, 1.0\}$, where the plan reward jointly accounts for structural validity and semantic alignment. Figure 15 reports the validation EM on Bamboogle evaluated every ten training steps. All settings exhibit steady improvements during the early stages of training, indicating that plan-level supervision provides effective guidance for multi-hop reasoning. Nevertheless, their later-stage behaviors differ noticeably. A smaller weight ($\lambda = \gamma = 0.25$) results in smoother but relatively slower performance gains, suggesting that weak plan supervision limits the exploitation of global reasoning signals. In contrast, a larger weight ($\lambda = \gamma = 1.0$) achieves competitive peak accuracy but shows clear instability toward the end of training, with abrupt performance degradation, implying that overly strong plan rewards may dominate optimization and reduce robustness.

The aggregated results across five benchmarks in Table 10 further support this observation. Overall, $\lambda = \gamma = 0.5$ achieves the most balanced performance, delivering consistent improvements on HotpotQA, 2WikiMultihopQA, and Bamboogle, while remaining competitive on Musique and WikiHop. Although $\lambda = \gamma = 1.0$ attains the best scores on some datasets, its advantages are less consistent and accompanied by unstable training dynamics. These results suggest that a moderate global-plan reward weight provides a favorable trade-off between effective plan guidance and stable optimization. Accordingly, we adopt $\lambda = \gamma = 0.5$ as the default configuration in our main experiments.

## K Transition Point Sensitivity Study

**Training Dynamics:** As illustrated in Figure 16, the reward curve for $T = 50$ exhibits a more rapid and stable ascent compared to $T = 100$. When $T = 100$, the model remains in the process-oriented learning phase for a longer duration, which, while helpful for initial planning, delays the optimization for final answer accuracy. In contrast, $T = 50$ provides a more balanced transition, allowing the model to quickly capitalize on refined reasoning paths to improve final outcomes.

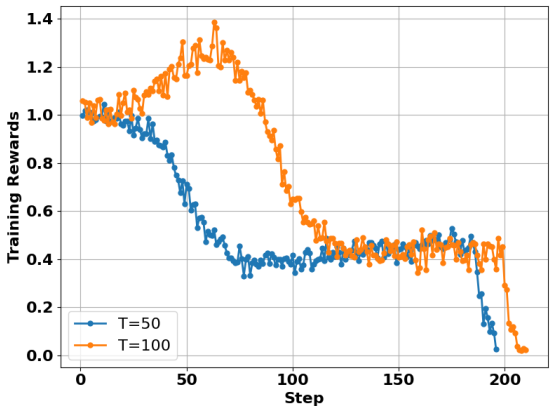



Figure 16: Training reward curves of the Qwen2.5-3B-Instruct model under different transition point settings (T=50 and T=100).

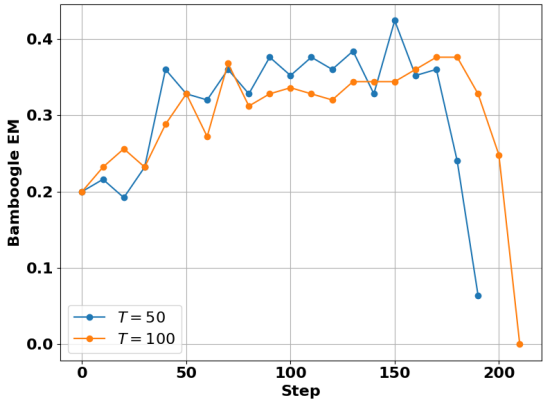



Figure 17: Performance comparison under transition point settings (T=50 and T=100) across five QA benchmarks, evaluated using Qwen2.5-3B-Instruct.

**Performance Comparison:** Table 11 and Figure 17 present the evaluation results. We observe that: $T = 50$ outperforms $T = 100$ on four out of five benchmarks (HotpotQA, 2WikiMultiHop, Musique, and WikiHop). The improvements in EM and F1 scores suggest that an earlier shift towards outcome-based optimization helps the model better internalize the relationship between global planning and correct answer generation.

Based on the empirical evidence, we conclude that $T = 50$ is the optimal transition point for GlobalRAG. It achieves the best trade-off between guiding the model's reasoning process and ensuring high-fidelity final answers. Setting the transition point too late (e.g., $T = 100$) may lead to over-optimization of intermediate steps at the expense

| Methods | HotpotQA | | 2Wiki | | Musique | | Bamboogle | | WikiHop | |
|---|---|---|---|---|---|---|---|---|---|---|
| | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| T=50 | 32.9 | 44.2 | 42.3 | 47.8 | 10.8 | 18.6 | 37.6 | 49.3 | 12.9 | 20.7 |
| T=100 | 31.2 | 43.1 | 40.6 | 46.6 | 10.0 | 17.0 | 39.2 | 50.6 | 12.2 | 19.8 |

Table 11: Performance comparison of Qwen2.5-3B-Instruct under different Transition Point $T$ across five QA benchmarks.

of final task performance. Consequently, $T = 50$ is adopted as the default setting for all other experiments in this study.

## L Detailed Results for Algorithmic Attribution Study

This appendix provides the complete quantitative results for the algorithmic attribution study summarized in the main text. All experiments reported here are conducted under strictly controlled training data settings, ensuring that both Search-R1 and GlobalRAG are trained on identical datasets. Additional experimental details and analysis can be found in the corresponding section of the main paper.

Table 3 reports EM and F1 scores across five multi-hop question answering benchmarks using three backbone models of different scales: Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, and Llama3.1-8B-Instruct. These results complement the main-text discussion by presenting the full benchmark-level breakdown.

Under all model configurations, GlobalRAG consistently achieves higher performance than Search-R1 across datasets. The improvements are especially evident on benchmarks that require complex multi-hop reasoning and evidence aggregation, such as HotpotQA, MuSiQue, and Bamboogle. Notably, the performance gap generally increases with model scale, indicating that GlobalRAG is able to more effectively exploit larger backbone capacity.

On MuSiQue, which emphasizes long-horizon reasoning and structured knowledge integration, GlobalRAG demonstrates relatively larger gains compared to Search-R1 across all backbones. This observation further supports the main-text claim that the proposed algorithmic design better captures global reasoning structure under challenging conditions.

Overall, the detailed results reported in this appendix provide comprehensive empirical support for the conclusion drawn in the main text: when trained on identical data, the observed performance improvements of GlobalRAG over Search-R1 are attributable to algorithmic differences rather than data-related factors.

## M 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, our global planning strategy enables more precise retrieval of relevant evidence. As shown in Table 12 and Table 13, GlobalRAG consistently retrieves the key information needed for complex queries, whereas Search-R1 often returns incomplete or less relevant content. Moreover, explicit `<subPlan>` steps and sub-answer validation rewards help filter distracting evidence and ensure the correctness of intermediate reasoning and sub-answers. This advantage is especially clear in Table 14 and Table 15, where GlobalRAG remains coherent and accurate even under misleading information, demonstrating improved reliability on complex QA 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 12: 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 13: 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 14: 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 15: 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**.