

# CloRe: Closed-Loop Reasoning via Intermediate-State Maintenance for Multi-Hop Question Answering

Yingxin Pei\*, Yuantao Han\*, Ruiting Li, Binquan Ji, Haibo Luo, Yifei Lu, Feiliang Ren†

School of Computer Science and Engineering,  
Northeastern University, Shenyang 110819, China  
peiyngxin668@gmail.com, renfeiliang@cse.neu.edu.cn

## Abstract

Retrieval-Augmented Generation (RAG) has substantially advanced knowledge-intensive question answering (QA). Existing methods typically follow a “decompose and sequentially solve” paradigm, mainly optimizing sub-question generation, retrieval, or search decisions, while lacking explicit maintenance of cross-step intermediate reasoning states. As a result, retrieval noise, query drift, and local reasoning errors introduced in early steps can be inherited and amplified along the reasoning chain, eventually causing reasoning failure. We propose **CloRe**, a state-based closed-loop reasoning framework for multi-hop question answering that explicitly maintains hop-level intermediate states and coordinates forward review with dependency-aware backward correction. In the forward phase, CloRe uses conditional query rewriting to mitigate query drift, stage-wise evidence acquisition to mitigate retrieval noise, and step-level review with local repair to filter unreliable intermediate answers before they are propagated. When forward correction is insufficient, CloRe enters a backward phase that uses downstream failure signals to backtrack along explicit dependencies and revise predecessor states that were locally acceptable under earlier review but become incompatible with later dependency constraints. Experiments on three representative multi-hop QA benchmarks show that CloRe achieves competitive performance against strong baselines, demonstrating the effectiveness of explicit intermediate-state maintenance for multi-hop reasoning.

## 1 Introduction

In recent years, large language models (LLMs) have demonstrated remarkable reasoning capabilities. For multi-hop questions, early methods commonly adopt prompt-based decomposition and step-wise reasoning, such as Least-to-Most (Zhou et al.,

\* Equal contribution. † Corresponding author.  
Code: <https://github.com/peiyxyx/CloRe>

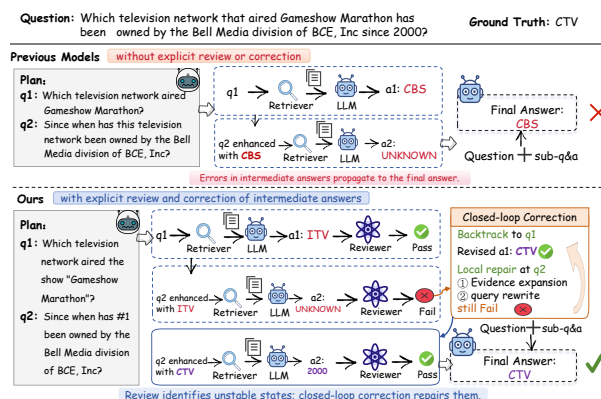


Figure 1: The process differences between our approach and the previous iterative generation framework.

2023), Plan-and-Solve (Wang et al., 2023), Self-Ask (Press et al., 2023), and Decomposed Prompting (Khot et al., 2023). However, these methods mainly rely on model’s parametric knowledge and prompt-induced intermediate reasoning processes.

To alleviate this issue, Retrieval-Augmented Generation (RAG) has been widely introduced into knowledge-intensive question answering, enabling models to access external evidence beyond parametric memory. However, conventional methods that rely only on fixed retrieval are often insufficient to provide the evidence required by subsequent reasoning steps. To better support multi-step reasoning with external evidence, subsequent studies, such as IRCoT (Trivedi et al., 2023), Search-in-the-Chain (Xu et al., 2024), GenGround (Shi et al., 2024), and DEC (Ji et al., 2025), integrate retrieval more tightly into multi-hop inference through iterative retrieval, forward refinement, or decomposition-based retrieval. Although some methods perform verification or refinement, they typically operate along the forward trajectory and focus on improving the current reasoning step. More recently, methods including Search-R1 (Jin et al., 2025), StepSearch (Zheng et al., 2025), GlobalRAG (Luo et al., 2025), and ReasonRAG (Zhang

et al., 2026) further improve retrieval or search decisions through reinforcement learning. However, these methods still primarily focus on improving retrieval, answer refinement, or search behavior at the current step, while providing limited explicit maintenance of cross-step intermediate states. As a result, retrieval noise and local reasoning errors introduced in early steps can be inherited by later hops, and downstream failure signals are not effectively used to determine whether earlier intermediate conclusions have become unreliable.

To address these limitations, we propose **CloRe**, a closed-loop reasoning framework for multi-hop question answering. As illustrated in Figure 1, CloRe organizes multi-hop solving as a dependency-aware reasoning process over hop-level intermediate states. In the forward phase, query rewriting resolves dependency placeholders and improves retrieval alignment; stage-wise evidence acquisition starts from compact evidence and expands only when the current context is insufficient, thereby reducing distraction from weakly relevant evidence; and step-level review with local repair reduces the chance that unreliable intermediate answers are directly committed to later hops. When forward correction is still insufficient, CloRe enters a backward phase that uses downstream failure signals to backtrack along explicit dependencies and revise predecessor states that were locally acceptable under earlier review but prove incompatible with later dependency constraints. In this way, CloRe not only improves current-hop retrieval and answer generation, but also supports explicit maintenance and correction of cross-step intermediate states throughout the reasoning process.

Our contributions are summarized as follows:

- We propose **CloRe**, a state-based closed-loop reasoning framework that explicitly maintains hop-level intermediate states and coordinates forward review with dependency-aware backward correction for multi-hop QA.
- We evaluate CloRe on multiple representative multi-hop QA benchmarks, demonstrating competitive performance against strong baselines.
- We further analyze CloRe’s inference behavior and efficiency, showing that explicit intermediate-state maintenance improves multi-hop reasoning while controlling the overhead of closed-loop correction.

## 2 Related work

**Step-by-Step Reasoning and Question Decomposition.** A representative line of work in multi-hop QA decomposes a complex question into intermediate sub-problems for step-wise solving. Early work such as Decomprc explicitly decomposes questions into single-hop sub-questions and solves multi-hop QA by reranking candidate decompositions (Min et al., 2019). With the rise of LLMs, methods such as Chain-of-Thought (Wei et al., 2022), Least-to-Most (Zhou et al., 2023), Decomposed Prompting (Khot et al., 2023), Plan-and-Solve (Wang et al., 2023), and Self-Ask (Press et al., 2023) demonstrate the effectiveness of explicitly generating intermediate reasoning steps for complex tasks. More recent work improves decomposition quality and iterative reasoning, for example, through guided decomposition or self-critique-based iterative refinement (Liu et al., 2025a; Chu et al., 2025). While these approaches effectively improve step-wise reasoning and decomposition quality, they mainly optimize intermediate generation or refinement within the forward reasoning process, without explicitly maintaining dependency-aware intermediate states across multiple hops. Consequently, early-step errors may propagate along the reasoning chain, as later reasoning signals cannot effectively trigger revisions of previous intermediate conclusions.

**Retrieval-Augmented Generation for Multi-Hop QA.** RAG combines parametric knowledge with external retrieval to support knowledge-intensive reasoning (Guu et al., 2020; Lewis et al., 2020; Izacard and Grave, 2021). For multi-hop QA, later work increasingly integrates retrieval into iterative reasoning, since the evidence required at one step often depends on what has already been established in previous steps (Trivedi et al., 2023). Representative methods such as IRCoT (Trivedi et al., 2023), GenGround (Shi et al., 2024), DEC (Ji et al., 2025), ChainRAG (Zhu et al., 2025), Q-DREAM (Ye et al., 2025), and HopRAG (Liu et al., 2025b) interleave retrieval with intermediate reasoning, query refinement, or structure-aware evidence selection to improve multi-hop evidence acquisition. More recent methods, including Search-R1 (Jin et al., 2025), StepSearch (Zheng et al., 2025), GlobalRAG (Luo et al., 2025), and ReasonRAG (Zhang et al., 2026), further optimize retrieval or search behaviors through reinforcement learning. However, these methods primarily focus on improving

retrieval decisions, evidence utilization, or search trajectories, while lacking explicit maintenance of dependency-aware intermediate states across multiple hops. In contrast, CloRe explicitly models intermediate answers as maintainable states and coordinates forward review with dependency-aware backward correction, enabling downstream failures to revise earlier states when they become incompatible with later reasoning constraints.

### 3 Methodology

#### 3.1 Problem Formulation

CloRe formulates multi-hop QA as a state-based closed-loop reasoning process, rather than a purely forward decomposition-and-solving process. As illustrated in Figure 2, the forward phase advances reasoning hop by hop under explicit dependencies, while the backward phase uses downstream failure signals to revise unreliable predecessor states, supporting dynamic maintenance of intermediate reasoning states.

**Problem Decomposition.** Given an input question  $x$ , a trained question planner (QP) decomposes it into a sequence of  $T$  sub-questions:

$$Y = \text{QP}(x) = \{(q_t, r_t)\}_{t=1}^T, \quad Q = \{q_t\}_{t=1}^T \quad (1)$$

where  $q_t$  denotes the  $t$ -th sub-question and  $r_t \in \{0, 1\}$  specifies whether external retrieval is required. To model cross-step dependencies via placeholders, we define the dependency index set  $\mathcal{D}_t \subseteq \{1, \dots, t-1\}$ , where  $k \in \mathcal{D}_t$  indicates that  $q_t$  references the answer of  $q_k$ .

**State Representation.** At step  $t$ , the execution state is defined as:

$$s_t = \{H_t, B_t\}, \quad H_0 = B_0 = \emptyset \quad (2)$$

where  $H_t$  denotes the committed history and  $B_t$  represents the exclusion memory. For each accepted hop, the committed history is updated as:

$$H_t = H_{t-1} \cup \{(t, q_t, \tilde{q}_t, \hat{a}_t)\} \quad (3)$$

This tuple stores the step index, original sub-question, rewritten query, and accepted answer.

To update hop  $t$ , CloRe uses the dependency-relevant subsets associated with  $q_t$  rather than the entire previous state:

$$H_{t-1}^{\text{dep}} = \{(i, q_i, \tilde{q}_i, \hat{a}_i) \in H_{t-1} \mid i \in \mathcal{D}_t\} \quad (4)$$

$$B_{t-1}^{\text{dep}} = \{(i, q_i, \tilde{q}_i, a_i^{\text{rej}}) \in B_{t-1} \mid i \in \mathcal{D}_t\} \quad (5)$$

**Committable States.** Committability is primarily defined for retrieval-required hops. Specifically, when  $r_t = 1$ , a candidate answer is committed only if it satisfies two conditions: (1) sufficient grounding in local evidence, and (2) consistency with the existing history  $H_{t-1}$ . When  $r_t = 0$ , the system directly generates the answer from the current sub-question and the relevant history.

**State-Based Hop Transition.** At hop  $t$ , the update depends on the current sub-question  $(q_t, r_t)$  and the dependency-relevant state from the previous step, formalized as:

$$s_t \sim \mathcal{T}(s_{t-1}^{\text{dep}}, (q_t, r_t)), \quad s_{t-1}^{\text{dep}} = \{H_{t-1}^{\text{dep}}, B_{t-1}^{\text{dep}}\} \quad (6)$$

where  $\mathcal{T}(\cdot)$  denotes the closed-loop hop-level transition defined by forward reasoning and, when triggered, backward correction.

#### 3.2 Closed-Loop Inference

##### 3.2.1 Problem Planning

To improve decomposition stability, we fine-tune a lightweight question planner based on Qwen3-4B-Instruct (Yang et al., 2025) to generate structured sub-question sequences for closed-loop inference. The planner is trained on data constructed from HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022) to produce structurally valid decompositions with explicit dependency references and retrieval indicators. Specifically, retrieval-free steps typically involve logical operations (e.g., substitution, comparison, counting, or aggregation) over historical sub-answers, whereas retrieval-required steps invoke external evidence acquisition. The planner is trained in two consecutive stages: Supervised Fine-Tuning (SFT) (Ouyang et al., 2022) followed by Direct Preference Optimization (DPO) (Rafailov et al., 2024). Detailed data construction, training objectives, implementation settings, and planner ablations are provided in Appendix B.2.

##### 3.2.2 Forward Reasoning

The forward phase proceeds hop by hop over the current sub-question  $(q_t, r_t)$  and prior state  $s_{t-1}$  through the following phases.

**Phase 1: Conditional Rewriting.** When the current sub-question contains placeholder references, it is conditionally rewritten using the relevant history subset  $H_{t-1}^{\text{dep}}$  to resolve dependency references:

$$\tilde{q}_t = W(q_t, H_{t-1}^{\text{dep}}), \quad (7)$$

where  $W(\cdot)$  denotes the query rewriter.

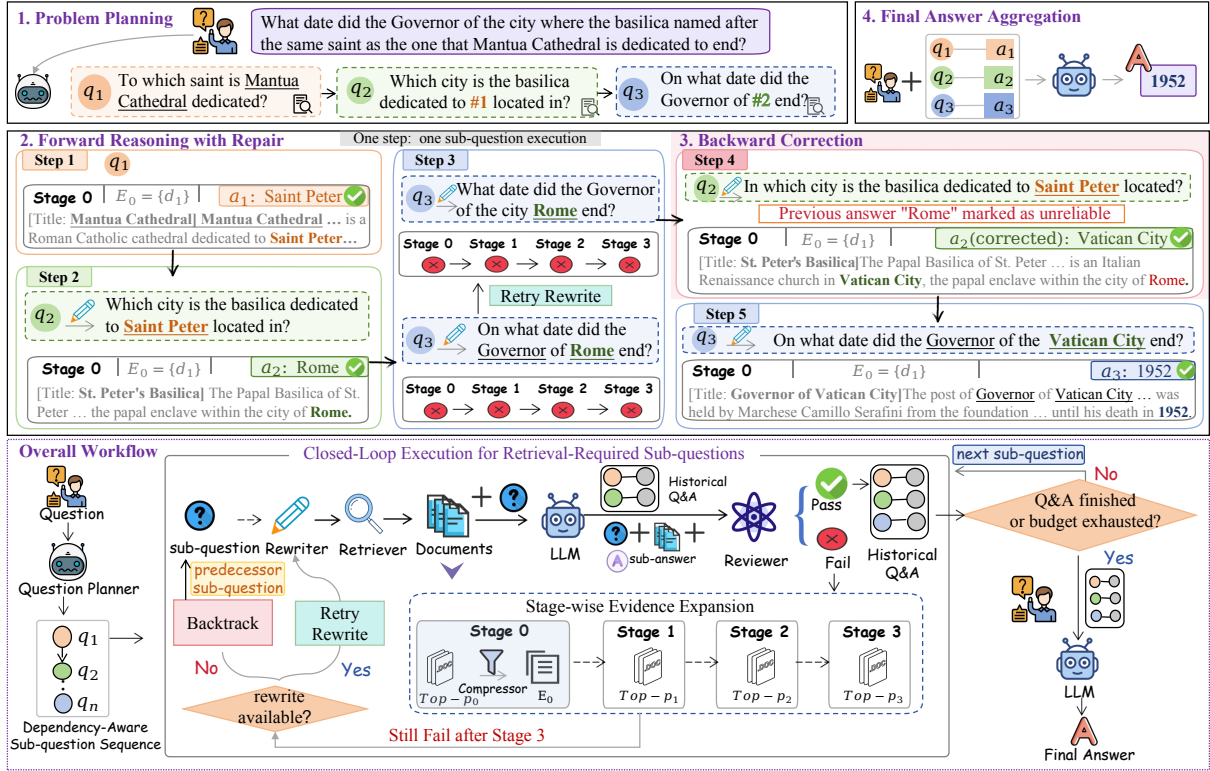


Figure 2: Overview of the CloRe framework. (a) Decompose the input question into a dependency-aware sub-question chain. (b) Execute forward reasoning sequentially, where each sub-question is conditionally rewritten when needed to resolve placeholder references, answered with stage-wise evidence, and reviewed with forward repair. (c) If a downstream sub-question still fails after all evidence stages and an additional rewrite, backtrack to its explicit predecessor and revise the affected reasoning state. (d) Resume forward reasoning from the corrected dependency path and aggregate the accepted sub-answers to produce the final answer.

**Phase 2: Stage-Wise Evidence Acquisition.** For retrieval-required hops ( $r_t = 1$ ), CloRe first retrieves an initial set of candidate documents  $\{d_t^m\}_{m=1}^M$  with relevance scores  $\{\alpha_t^m\}_{m=1}^M$ , normalizing them into a probability distribution:

$$\pi_t = \text{softmax}(\alpha_t). \quad (8)$$

To reduce distraction from weakly relevant passages and lower the risk that key supporting facts are diluted, CloRe begins with a compact Stage-0 evidence set. Under an increasing top- $p$  schedule  $p_0 \leq p_1 \leq \dots \leq p_K$  ( $p_K = 1$ ), the framework constructs stage-wise local evidence  $E_t^{(j)}$  across stages  $j \in \{0, \dots, K\}$ :

$$E_t^{(j)} = \begin{cases} \Psi(D_t^{(0)}, \tilde{q}_t), & j = 0, \\ D_t^{(j)}, & j = 1, \dots, K, \end{cases} \quad (9)$$

where  $\Psi(\cdot)$  denotes a query-aware compressor that prunes the initial subset  $D_t^{(0)}$  to retain only documents matching salient query units in  $\tilde{q}_t$ . If a candidate answer fails review, the evidence scope ex-

pands incrementally, introducing additional cross-document support only when the current evidence subset is insufficient (see Appendix C for details).

**Phase 3: Candidate Answer Generation and Review.** At stage  $j$ , the candidate answer  $\hat{a}_t^{(j)}$  is generated as follows:

$$\hat{a}_t^{(j)} = \begin{cases} G(\tilde{q}_t, E_t^{(j)}), & r_t = 1, \\ G(\tilde{q}_t, H_{t-1}^{\text{dep}}), & r_t = 0, \end{cases} \quad (10)$$

where  $G(\cdot)$  denotes the answer generator. For retrieval-required hops, the candidate answer is evaluated by a step-level reviewer:

$$o_t^{(j)} = \mathcal{A}(\tilde{q}_t, \hat{a}_t^{(j)}, E_t^{(j)}, H_{t-1}^{\text{dep}}), \quad (11)$$

$$o_t^{(j)} \in \{\text{accept}, \text{revise}, \text{reject}\},$$

where  $\mathcal{A}(\cdot)$  denotes the reviewer and  $o_t^{(j)}$  is the review outcome. An accepted or revised answer is then committed to history  $H_t$ .

**Phase 4: Local Repair.** If the candidate answer is rejected, CloRe first attempts local repair within

the forward loop. Let  $\rho_t$  denote the number of retry rewrites already used at hop  $t$ , and let  $R_{\max}$  be the rewrite budget. The repair action is determined as:

$$u_t^{(j)} = \begin{cases} \text{Expand,} & j < K, \\ \text{Rewrite,} & j = K, \rho_t < R_{\max}, \\ \text{Backtrack,} & \text{otherwise,} \end{cases} \quad (12)$$

where Expand advances to the next evidence stage, prioritizing additional evidence before altering the query; Rewrite performs failure-driven rewriting of the current sub-question to enrich underspecified semantics or improve retrieval alignment; and Backtrack invokes backward correction once all evidence stages have been explored and the rewrite budget is exhausted.

### 3.2.3 Backward Correction

When the forward phase triggers  $u_t^{(j)} = \text{Backtrack}$ , CloRe performs backward correction along explicit dependencies, suggesting that a previously accepted predecessor state, although locally valid under initial verification, may be insufficient for subsequent downstream reasoning.

**Direct Dependency Backtracking.** Backward correction is restricted to the explicit predecessor set of the current sub-question. When multiple predecessors exist, CloRe selects the most recent direct predecessor as the backtracking target, which prioritizes correcting the closest upstream state while avoiding unnecessary recomputation. When backtracking is triggered, CloRe returns to a predecessor hop  $k_t^* \in \mathcal{D}_t$  and updates the exclusion memory:

$$B_t = B_{t-1} \cup \{(k_t^*, q_{k_t^*}, \tilde{q}_{k_t^*}, a_{k_t^*}^{\text{rej}})\}, \quad (13)$$

where  $a_{k_t^*}^{\text{rej}}$  denotes the previously accepted answer invalidated by downstream failure. If  $\mathcal{D}_t = \emptyset$ , backward correction is not applied.

**Rollback and Re-solving.** After the memory update, CloRe performs a local rollback on the committed history and resumes forward reasoning from hop  $k_t^*$ :

$$H_t = \text{Rollback}(H_{t-1}, k_t^*), \quad (14)$$

where  $\text{Rollback}(\cdot)$  removes affected committed records from hop  $k_t^*$  onward while preserving the valid prefix. CloRe re-enters the forward phase with the updated state  $(H_t, B_t)$  and sequentially re-solves the affected downstream sub-questions.

### 3.2.4 Final Answer Aggregation

Inference alternates between forward reasoning and backward correction until either all sub-questions have been resolved with committable answers and no backtracking is triggered, or the predefined global inference budget is exhausted. Let  $H_\tau$  denote the committed history at termination. The final answer is generated as:

$$\hat{y} = G_{\text{agg}}(x, H_\tau), \quad (15)$$

where  $G_{\text{agg}}(\cdot)$  denotes the answer aggregator. Prompt templates are provided in Appendix G.

## 4 Experiments

### 4.1 Datasets and Evaluation Metrics

**Datasets.** We evaluate on three multi-hop QA benchmarks: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022), which involve multi-step reasoning over evidence from multiple documents (see Appendix A.1 for more details). Following the evaluation settings of IR-CoT (Trivedi et al., 2023) and DEC (Ji et al., 2025), which use 500 validation examples per benchmark, we further expand the evaluation scale by sampling 2,500 validation examples from HotpotQA and 2WikiMultiHopQA with a fixed random seed (seed=42), and using the full validation set of MuSiQue containing 2,417 examples. All methods are evaluated on identical examples.

**Evaluation Metrics.** We adopt Exact Match (EM) and word-level F1 as the evaluation metrics. Details are provided in Appendix A.2.

### 4.2 Baseline Models

We compare CloRe against representative baselines from both non-RL and search-based RL methods:

(1) **Non-RL baselines:** NaiveRAG (Lewis et al., 2020), IRCOT (Trivedi et al., 2023), Self-Ask (Press et al., 2023), GenGround (Shi et al., 2024), and DEC (Ji et al., 2025).

(2) **Search-RL baselines:** Search-R1 (Jin et al., 2025), StepSearch (Zheng et al., 2025), Reason-RAG (Zhang et al., 2026), and GlobalRAG (Luo et al., 2025) (see details in Appendix A.3).

### 4.3 Implementation Details

We conduct experiments on the open-source model Qwen2.5-7B-Instruct (Yang et al., 2024) to assess the effectiveness of our method across different backbones.

| Method       | HotpotQA               |                           | 2WikiMultiHopQA        |                           | MuSiQue                |                           |
|--------------|------------------------|---------------------------|------------------------|---------------------------|------------------------|---------------------------|
|              | EM (95% CI)            | F1 (95% CI)               | EM (95% CI)            | F1 (95% CI)               | EM (95% CI)            | F1 (95% CI)               |
| NaiveRAG     | 43.6[41.7,45.6]        | 52.38[50.59,54.16]        | 29.4[27.6,31.2]        | 37.29[35.51,38.99]        | 14.2[12.9,15.6]        | 20.61[19.16,22.06]        |
| IRCoT        | 47.2[45.3,49.2]        | 58.54[56.82,60.27]        | 38.2[36.4,40.1]        | 50.66[48.94,52.46]        | 17.3[15.8,18.8]        | 23.98[22.44,25.56]        |
| Self-Ask     | 46.8[44.8,48.7]        | 54.68[52.85,56.51]        | 52.8[50.9,54.8]        | 60.39[58.60,62.19]        | 24.6[22.8,26.3]        | 32.50[30.74,34.20]        |
| GenGround    | 46.1[44.1,48.0]        | 55.59[53.79,57.34]        | 45.6[43.6,47.6]        | 58.25[56.52,60.01]        | 19.1[17.6,20.6]        | 29.24[27.70,30.87]        |
| DEC          | 49.0[47.0,51.0]        | 57.85[56.08,59.68]        | <u>55.6[53.7,57.6]</u> | <u>64.04[62.27,65.78]</u> | 24.1[22.4,25.8]        | 31.49[29.82,33.20]        |
| Search-R1    | 55.3[53.4,57.3]        | 63.10[61.35,64.85]        | 53.5[51.5,55.4]        | 61.13[59.36,62.90]        | 28.2[26.4,30.0]        | 35.98[34.24,37.75]        |
| StepSearch   | 54.5[52.5,56.4]        | 61.23[59.43,63.06]        | 49.7[47.7,51.6]        | 56.98[55.17,58.77]        | 29.6[27.8,31.4]        | 36.27[34.49,38.07]        |
| ReasonRAG    | 48.2[46.3,50.2]        | 55.69[53.92,57.52]        | 46.6[44.7,48.6]        | 53.71[51.90,55.56]        | 20.9[19.3,22.5]        | 28.00[26.34,29.68]        |
| GlobalRAG    | 49.5[47.6,51.4]        | 57.01[55.16,58.82]        | 49.5[47.5,51.4]        | 55.58[53.76,57.40]        | 23.2[21.5,24.9]        | 30.14[28.44,31.83]        |
| <b>CloRe</b> | <b>57.2[55.3,59.2]</b> | <b>63.83[62.02,65.60]</b> | <b>63.8[62.0,65.7]</b> | <b>71.80[70.17,73.44]</b> | <b>31.4[29.5,33.3]</b> | <b>37.96[36.09,39.80]</b> |

Table 1: Main results on three multi-hop QA benchmarks using Qwen2.5-7B-Instruct. Each entry reports the score with 95% bootstrap confidence intervals. Best results are in bold and second-best are underline.

For the retrieval corpus, HotpotQA and 2WikiMultiHopQA utilize the Wikipedia snapshots provided by the dataset creators. Since MuSiQue lacks an officially curated knowledge base, we follow prior work and use the 2020 Wikipedia version from 2WikiMultiHopQA (Ji et al., 2025). The retrieval process consists of two stages. We first use E5-base (Wang et al., 2022) for coarse retrieval and keep the top 45 passages, followed by bge-reranker-v2-m3 (Chen et al., 2024b) for fine-grained reranking to obtain the top 10 candidate passages. All methods are given access to the same reranked top-10 candidate pool. Following their original settings, all baselines except DEC use the top 3 reranked passages, whereas DEC selects the two most relevant documents from the top 10 together with documents that match specific keywords.

#### 4.4 Main Results

Table 1 compares CloRe with competitive non-RL and search-RL baselines across three multi-hop QA benchmarks, yielding the following observations:

**(1) Competitive performance across multi-hop QA benchmarks.** CloRe achieves the best point performance on all three benchmarks, improving EM over the strongest baseline by 1.9%, 8.2%, and 1.8% on HotpotQA, 2WikiMultiHopQA, and MuSiQue, respectively.

**(2) Gains vary across benchmarks.** CloRe achieves its largest gain on 2WikiMultiHopQA (+8.2% EM over DEC; 95% CI [6.52, 9.92]), where multi-step dependency tracking is more challenging. Gains on HotpotQA and MuSiQue remain positive but inconclusive, with corresponding CIs of [-0.04, 3.84] and [-0.12, 3.89] for EM gains over their strongest baselines, respectively. This suggests that intermediate-state maintenance is more

beneficial under stronger cross-hop dependencies.

**(3) Closed-Loop Intermediate-State Maintenance Mitigates Error Propagation.** Compared with forward decomposition and retrieval-based methods, CloRe achieves average EM gains of +9.4%, +7.9%, and +10.07% over Self-Ask, DEC, and GlobalRAG, respectively. While DEC mitigates retrieval noise through keyword filtering and GlobalRAG emphasizes sub-goal planning, these methods mainly follow forward execution and may accumulate errors across hops. In contrast, CloRe reviews intermediate answers before commitment and corrects unreliable states when necessary, reducing error propagation along dependency chains.

Compared with GenGround’s iterative forward refinement, CloRe improves EM by 11.1%, 18.2%, and 12.3% on HotpotQA, 2WikiMultiHopQA, and MuSiQue, respectively. These results suggest that maintaining and revising intermediate states across reasoning steps provides better cross-step consistency than local refinement alone.

#### 4.5 Efficiency Analysis

We report mean end-to-end latency per question on the expanded evaluation sets in Section 4.4. Non-RL methods use the same backbone, hardware, and serving setup, while search-RL methods are excluded due to incompatible interaction loops and serving architectures.

**Compared with simpler forward-only methods.** NaiveRAG, IRCoT, and Self-Ask are faster but substantially less accurate, while CloRe offers a more favorable accuracy–latency trade-off.

**Compared with decomposition-based DEC.** Although CloRe adds step-level review and dependency-aware backtracking, it reduces latency by 43.0% on HotpotQA and 31.6% on 2WikiMulti-

| Method             | HotpotQA      |               | 2WikiMultiHopQA |               | MuSiQue        |                |
|--------------------|---------------|---------------|-----------------|---------------|----------------|----------------|
|                    | EM            | F1            | EM              | F1            | EM             | F1             |
| <b>CloRe(Ours)</b> | <b>57.20</b>  | <b>64.26</b>  | <b>63.70</b>    | <b>71.03</b>  | <b>34.80</b>   | <b>41.57</b>   |
| w/o CRC            | 52.40 (↓8.4%) | 58.86 (↓8.4%) | 59.60 (↓6.4%)   | 65.11 (↓8.3%) | 25.20 (↓27.6%) | 29.25 (↓29.6%) |
| w/o FE             | 52.60 (↓8.0%) | 62.69 (↓2.4%) | 59.80 (↓6.1%)   | 66.80 (↓6.0%) | 27.20 (↓21.8%) | 34.07 (↓18.0%) |
| w/o QR             | 56.60 (↓1.0%) | 63.40 (↓1.3%) | 61.20 (↓3.9%)   | 68.47 (↓3.6%) | 28.60 (↓17.8%) | 35.64 (↓14.3%) |
| w/o BT             | 55.20 (↓3.5%) | 61.50 (↓4.3%) | 63.20 (↓0.8%)   | 70.32 (↓1.0%) | 31.00 (↓10.9%) | 37.47 (↓9.9%)  |

Table 2: Results of the ablation study on Qwen2.5-7B-Instruct. CRC, FE, QR, and BT denote Closed-loop review and correction, forward evolution, failure-driven query rewriting, and backtracking, respectively.

| Method    | HotpotQA | 2WikiMultiHopQA | MuSiQue |
|-----------|----------|-----------------|---------|
| NaiveRAG  | 2.52     | 3.48            | 1.40    |
| IRCoT     | 7.31     | 11.00           | 3.96    |
| Self-Ask  | 5.08     | 7.68            | 3.52    |
| GenGround | 32.23    | 31.27           | 24.37   |
| DEC       | 17.95    | 20.85           | 13.54   |
| CloRe     | 10.23    | 14.26           | 15.36   |

Table 3: Mean end-to-end latency (seconds) per question on three benchmarks.

HopQA, consistent with its shallow-first evidence expansion. On MuSiQue, CloRe is 13.4% slower, as complex dependencies more frequently trigger evidence expansion, rewriting, and backtracking.

**Compared with forward-refinement-based GenGround.** CloRe reduces latency by 68.3%, 54.4%, and 37.0% on HotpotQA, 2WikiMultiHopQA, and MuSiQue, respectively. GenGround repeatedly performs generation, grounding, and refinement, whereas CloRe selectively expands evidence and deepens correction only for unresolved cases.

CloRe controls closed-loop correction overhead and offers a competitive accuracy–latency trade-off, although latency increases on dependency-intensive benchmarks. Additional efficiency statistics are reported in Appendix D.

## 5 Further Analyses

All analyses in this section are conducted on 500 randomly sampled validation examples from each benchmark under a consistent evaluation setting.

### 5.1 Ablation Study

To examine the contribution of explicit cross-step intermediate-state maintenance in CloRe, we conduct module-level ablations on Qwen2.5-7B-Instruct. The results are reported in Table 2, from which we draw the following observations:

**(1) Impact of Intermediate-State Review and Correction.** Removing the step-level review-

and-correction mechanism reduces the system to sequential forward execution, where each sub-question is answered once using only the initial Stage-0 evidence. Under this setting, performance degrades substantially across all three datasets (6.4%–27.6% in EM). These results suggest that sequential forward execution alone is insufficient for stable multi-hop reasoning: without explicit review and correction, unreliable intermediate answers are more likely to be committed and subsequently propagated along dependency chains to later steps.

**(2) Impact of Forward Repair Mechanisms.** Disabling forward evolution (w/o FE) consistently reduces accuracy across all three datasets, with the largest drop on MuSiQue (EM -21.8%), while removing failure-driven query rewriting (w/o QR) also hurts performance, again most notably on MuSiQue (EM -17.8%). These results suggest that stage-wise evidence expansion and query rewriting play complementary roles within the forward loop: the former mitigates insufficient evidence at the current hop, while the latter helps improve retrieval alignment. Removing either component weakens the system’s ability to maintain reliable intermediate states before they are propagated downstream.

**(3) Impact of Backtracking Mechanism.** When backtracking is removed (w/o BT), performance drops slightly on HotpotQA and 2WikiMultiHopQA, but more noticeably on MuSiQue (EM -10.9%). This suggests that for instances with simpler dependency chains that can be resolved within the current hop, forward repair is often sufficient. In contrast, for cases with stronger cross-step dependencies, once an incorrect intermediate state has been committed, local repair alone may be insufficient. In such cases, backtracking is useful when a previously accepted answer is insufficiently specific, weakly grounded, or incompatible with downstream dependency constraints, allowing the predecessor state to be revised and reasoning to

resume along a corrected dependency path.

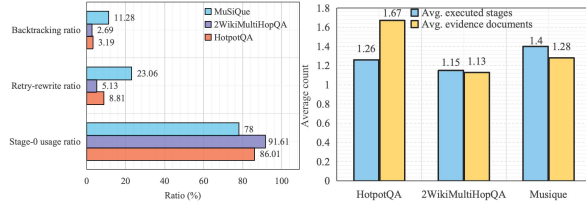


Figure 3: Execution behavior of the full CloRe system on retrieval-required sub-questions. Left: Stage-0 usage ratio, retry-rewrite ratio, and backtracking ratio. Right: average number of executed stages and average number of evidence documents per retrieval sub-question.

## 5.2 Execution Behavior of the Full System

Figure 3 shows two clear patterns in the execution behavior of the full CloRe system.

**(1) The full system follows a shallow-first and low-overhead execution pattern.** The left panel shows that the Stage-0 usage ratio reaches 86.01%, 91.61%, and 78.00% on HotpotQA, 2WikiMultiHopQA, and MuSiQue, respectively, indicating that most retrieval-required sub-questions are resolved at the earliest stage. Consistently, the right panel shows that the average number of executed stages remains close to 1 (1.26, 1.15, and 1.40), while the average number of evidence documents also stays low (1.67, 1.13, and 1.28). Taken together, these results suggest that the gains of CloRe are not obtained by pushing most instances through the full expansion schedule or by substantially increasing evidence volume.

**(2) More challenging benchmarks trigger the full correction pipeline more often.** MuSiQue exhibits higher retry-rewrite ratio (23.06%) and backtracking ratio (11.28%) compared to HotpotQA and 2WikiMultiHopQA, together with the highest average execution depth (1.40) in the right panel. This pattern suggests that, although the full system usually remains shallow, more challenging cases rely more heavily on the complete correction pipeline.

## 5.3 Evolution Depth Study

To examine the empirical effect of the evidence-evolution mechanism, we vary the evolution depth  $k$  while disabling query rewriting and backtracking correction. The configurations  $k \in \{0, 1, 2, 3\}$  correspond to progressively deeper evidence expansion, ranging from the initial compact evidence pool to the full evolution schedule [0.3, 0.9, 0.99, 1.0] (see Appendix E for details).

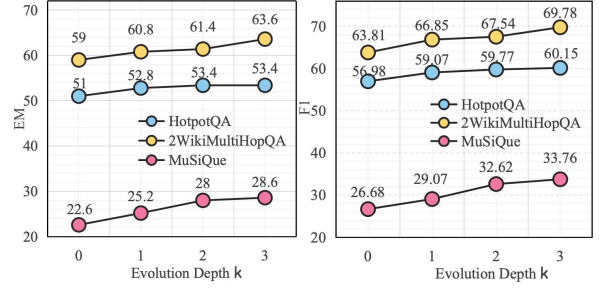


Figure 4: Performance under different evolution depths  $k$ . Left: EM on HotpotQA, 2WikiMultiHopQA, and MuSiQue. Right: F1 on the same datasets.

| Rewrite Budget | HotpotQA     |              | 2Wiki        |              | MuSiQue      |              |
|----------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                | EM           | F1           | EM           | F1           | EM           | F1           |
| 0              | 53.40        | 60.15        | <b>63.60</b> | 69.78        | 28.60        | 33.76        |
| 1              | <b>55.20</b> | <b>61.50</b> | 63.20        | <b>70.32</b> | 31.00        | <b>37.47</b> |
| 2              | 54.20        | 60.95        | 63.00        | 70.03        | <b>31.40</b> | 37.22        |
| 3              | 53.80        | 60.27        | 63.20        | 69.99        | 30.60        | 35.93        |

Table 4: Performance under different rewrite budgets with local backtracking disabled.

As illustrated in Figure 4, both EM and F1 improve monotonically with the evolution depth  $k$ . This trend suggests that compact early-stage evidence can already provide useful support, while deeper stages provide additional evidence for cases that remain underdetermined at shallow depths. Notably, the gains are already visible from  $k = 0$  to  $k = 1$  across all three benchmarks, indicating that even limited early expansion can recover useful missing evidence. Further increases from deeper stages remain beneficial, although the magnitude of the gains varies across datasets. In particular, 2WikiMultiHopQA and MuSiQue show larger improvements as  $k$  increases, suggesting that these benchmarks benefit more from broader evidence coverage in this controlled setting.

## 5.4 Rewrite Budget Study

To analyze the sensitivity of failure-driven query rewriting to the rewrite budget, we vary the budget under the fixed evidence-evolution schedule [0.3, 0.9, 0.99, 1.0] with backtracking disabled. Table 4 shows that a budget of 1 provides the most balanced overall setting across benchmarks, achieving the best performance on HotpotQA, the highest F1 on 2WikiMultiHopQA, and clear improvements over the no-rewrite setting on MuSiQue, whereas larger budgets lead to mixed or diminishing gains. This pattern suggests that one rewrite already captures much of the benefit of query reformulation,

| Dataset  | Reviewer        | EM    | F1    | Latency (s) |
|----------|-----------------|-------|-------|-------------|
| HotpotQA | 3B              | 57.40 | 64.56 | 8.97        |
|          | 7B <sup>†</sup> | 57.20 | 64.26 | 9.47        |
|          | 14B             | 60.40 | 67.68 | 12.63       |
|          | 32B             | 63.20 | 69.97 | 16.22       |
| 2Wiki    | 3B              | 61.60 | 68.10 | 13.79       |
|          | 7B <sup>†</sup> | 63.70 | 71.03 | 13.88       |
|          | 14B             | 65.40 | 73.07 | 16.03       |
|          | 32B             | 67.20 | 74.37 | 32.04       |
| MuSiQue  | 3B              | 30.80 | 36.61 | 12.38       |
|          | 7B <sup>†</sup> | 34.80 | 41.57 | 15.57       |
|          | 14B             | 36.60 | 43.39 | 19.02       |
|          | 32B             | 36.00 | 42.55 | 26.30       |

Table 5: Sensitivity to reviewer capacity. <sup>†</sup> marks the default reviewer.

| Metric            | HotpotQA | 2Wiki | MuSiQue |
|-------------------|----------|-------|---------|
| Macro-F1          | 76.00    | 71.86 | 75.35   |
| Cohen’s $\kappa$  | 0.673    | 0.616 | 0.650   |
| Error Recall (%)  | 76.47    | 75.00 | 78.06   |
| Recovery Acc. (%) | 85.26    | 87.04 | 84.64   |

Table 6: Human evaluation of 7B reviewer. Recovery Acc. denotes conditional recovery-selection accuracy.

while repeated reformulations may introduce additional semantic noise. We therefore set the failure-driven rewrite budget to 1 in the final system.

## 5.5 Reviewer Analysis

**Reviewer capacity.** In all experiments, the reviewer uses the same model as the backbone reasoner, i.e., Qwen2.5-7B-Instruct, with temperature 0.1. To examine sensitivity to reviewer capacity, we vary only the reviewer model. As shown in Table 5, larger reviewers generally improve EM/F1 at the cost of higher latency. We therefore use the 7B reviewer by default, matching the reasoner capacity without relying on a stronger auxiliary model.

**Reviewer stability.** We further evaluate the stability of the *accept/revise/reject* decisions by repeating each review case three times at the same temperature of 0.1. Across the three benchmarks,

| Methods   | HotpotQA     |              | 2Wiki        |              | MuSiQue      |              |
|-----------|--------------|--------------|--------------|--------------|--------------|--------------|
|           | EM           | F1           | EM           | F1           | EM           | F1           |
| NaiveRAG  | 45.60        | 53.53        | 30.20        | 35.70        | 13.80        | 19.18        |
| IRCOT     | 50.00        | 56.52        | 45.40        | 53.18        | 25.20        | 29.92        |
| Self-Ask  | 46.80        | 54.83        | 44.60        | 50.68        | 20.20        | 25.95        |
| GenGround | 41.40        | 49.28        | 39.40        | 46.46        | 14.60        | 21.21        |
| DEC       | 46.00        | 54.38        | 47.60        | 54.99        | 27.40        | 35.78        |
| Ours      | <b>53.00</b> | <b>61.60</b> | <b>59.40</b> | <b>67.11</b> | <b>34.60</b> | <b>41.40</b> |

Table 7: Results on Llama3.1-8B-Instruct.

| Methods   | HotpotQA     |              | 2Wiki        |              | MuSiQue      |              |
|-----------|--------------|--------------|--------------|--------------|--------------|--------------|
|           | EM           | F1           | EM           | F1           | EM           | F1           |
| NaiveRAG  | 56.80        | 66.34        | 54.60        | 64.24        | 28.40        | 39.09        |
| IRCOT     | 62.80        | 72.50        | 57.80        | 69.60        | 35.10        | 44.50        |
| Self-Ask  | 62.60        | 71.53        | 70.40        | 79.27        | 40.40        | 50.59        |
| GenGround | 63.80        | 72.61        | 60.40        | 72.70        | 36.20        | 46.27        |
| DEC       | 58.80        | 71.70        | 67.00        | 80.22        | 38.50        | 50.27        |
| Ours      | <b>64.00</b> | <b>73.73</b> | <b>71.20</b> | <b>81.58</b> | <b>44.40</b> | <b>55.03</b> |

Table 8: Results on GPT-5.1.

Fleiss’  $\kappa$  ranges from 0.9791 to 0.9901, with disagreement rates below 1.4%, indicating highly stable reviewer decisions under our inference setting. Detailed results are reported in Appendix E.5.

**Reviewer correctness.** We further evaluate the default 7B reviewer against human annotations on 748, 829, and 908 step-level cases from HotpotQA, 2WikiMultiHopQA, and MuSiQue, respectively. As shown in Table 6, the reviewer achieves macro-F1 scores of 71.86–76.00 and Cohen’s  $\kappa$  values of 0.616–0.673. It detects 75.00–78.06% of incorrect intermediate answers and selects the correct recovery action for 84.64–87.04% of detected errors. Detailed annotation protocols and class-wise results are provided in Appendix E.6.

## 5.6 Cross-Backbone Generalization

We further evaluate CloRe on Llama3.1-8B-Instruct (Dubey et al., 2024) and GPT-5.1 (OpenAI, 2025). Tables 7 and 8 show that CloRe achieves the best EM and F1 scores across all three benchmarks on both backbones.

## 6 Conclusion

In this work, we propose CloRe, a closed-loop reasoning framework for multi-hop question answering that explicitly maintains and evolves hop-level intermediate states during multi-hop reasoning. In the forward phase, reasoning proceeds through conditional query rewriting, stage-wise evidence acquisition, answer generation, step-level review, and local repair. In the backward phase, downstream failure signals trigger backtracking along explicit dependencies to revise predecessor states that become unreliable under later reasoning constraints. Experiments on multiple multi-hop QA benchmarks show that CloRe improves performance and remains effective across different backbone models.

## Limitations

Although CloRe demonstrates encouraging results, it still has several limitations:

The current framework is constrained by the quality of the retrieval candidate set, especially after reranking. Since later evidence expansion operates over the top 10 passages retained after E5 retrieval and bge-reranker-v2-m3 reranking, relevant evidence omitted at this stage cannot be recovered by subsequent expansion, which may limit downstream correction.

CloRe is also constrained by the quality of the planner-generated dependency structure. Since both conditional rewriting and backward correction operate over explicit sub-question dependencies, errors or omissions in the initial decomposition can misdirect later rewriting and backtracking, reducing the effectiveness of subsequent recovery.

## Acknowledgments

This work is funded by the National Natural Science Foundation of China (No.62576085).

## References

- Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024a. [Humans or LLMs as the judge? a study on judgement bias](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8301–8327, Miami, Florida, USA. Association for Computational Linguistics.
- Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024b. [M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.
- Zheng Chu, Huiming Fan, Jingchang Chen, Qianyu Wang, Mingda Yang, Jiafeng Liang, Zhongjie Wang, Hao Li, Guo Tang, Ming Liu, and Bing Qin. 2025. [Self-critique guided iterative reasoning for multi-hop question answering](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2415–2438, Vienna, Austria. Association for Computational Linguistics.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 510 others. 2024. [The llama 3 herd of models](#).
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. [Realm: retrieval-augmented language model pre-training](#). ICML’20.
- 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.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Gautier Izacard and Edouard Grave. 2021. [Leveraging passage retrieval with generative models for open domain question answering](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.
- Binqian Ji, Haibo Luo, Yifei Lu, Lei Hei, Jiaqi Wang, Tingjing Liao, Lingyu Wang, Shichao Wang, and Feiliang Ren. 2025. [Resource-friendly dynamic enhancement chain for multi-hop question answering](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16461–16479, Vienna, Austria. Association for Computational Linguistics.
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Serkan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. [Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning](#). In *Second Conference on Language Modeling*.
- Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. [Decomposed prompting: A modular approach for solving complex tasks](#). In *The Eleventh International Conference on Learning Representations*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with pagedattention](#). In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP ’23*, page 611–626, New York, NY, USA. Association for Computing Machinery.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

- Hao Liu, Zhengren Wang, Xi Chen, Zhiyu Li, Feiyu Xiong, Qinhan Yu, and Wentao Zhang. 2025a. [HoP-RAG: Multi-hop reasoning for logic-aware retrieval-augmented generation](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1897–1913, Vienna, Austria. Association for Computational Linguistics.
- Yi Liu, Xiangrong Zhu, Xiangyu Liu, Wei Wei, and Wei Hu. 2025b. [Avoiding knowledge edit skipping in multi-hop question answering with guided decomposition](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 16256–16272, Suzhou, China. Association for Computational Linguistics.
- Jinchang Luo, Ming-Ming Cheng, Fanqi Wan, Ni Li, Xiaoling Xia, Shuangshuang Tian, Tingcheng Bian, Haiwei Wang, Haohuan Fu, and Yan Tao. 2025. [Globalrag: Enhancing global reasoning in multi-hop question answering via reinforcement learning](#). *ArXiv*, abs/2510.20548.
- Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. [Multi-hop reading comprehension through question decomposition and rescoring](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6097–6109, Florence, Italy. Association for Computational Linguistics.
- OpenAI. 2025. Gpt-5.1 instant and gpt-5.1 thinking system card. OpenAI system card. Published November 12, 2025.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.
- 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.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.
- 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.
- 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.
- Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. [Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.
- Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2024. [Search-in-the-chain: Interactively enhancing large language models with search for knowledge-intensive tasks](#). In *Proceedings of the ACM Web Conference 2024, WWW ’24*, page 1362–1373, New York, NY, USA. Association for Computing Machinery.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 Technical Report](#). *arXiv e-prints*, arXiv:2505.09388.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Dayiheng Zhou, Guohao Fan, Haoran Huang, Junting Huang, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.
- 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.

Linhao Ye, Lang Yu, Zhikai Lei, Qin Chen, Jie Zhou, and Liang He. 2025. [Optimizing question semantic space for dynamic retrieval-augmented multi-hop question answering](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17814–17824, Vienna, Austria. Association for Computational Linguistics.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. [Rankrag: Unifying context ranking with retrieval-augmented generation in llms](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 121156–121184. Curran Associates, Inc.

Wenlin Zhang, Xiangyang Li, Kuicai Dong, Yichao Wang, Pengyue Jia, Xiaopeng Li, Yingyi Zhang, Derong Xu, Zhaocheng Du, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. 2026. [Process vs. outcome reward: Which is better for agentic RAG reinforcement learning](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Xuhui Zheng, Kang An, Ziliang Wang, Yuhang Wang, and Yichao Wu. 2025. [StepSearch: Igniting LLMs search ability via step-wise proximal policy optimization](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21805–21830, Suzhou, China. Association for Computational Linguistics.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyang Luo. 2024. [LlamaFactory: Unified efficient fine-tuning of 100+ language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. [Least-to-most prompting enables complex reasoning in large language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

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

## A Experimental Details

### A.1 Datasets

We evaluate our approach on three representative multi-hop question answering benchmarks: HotpotQA, 2WikiMultiHopQA, and MuSiQue, all of which are widely adopted to assess complex evidence aggregation and multi-step reasoning. Following standard protocols, our evaluation focuses on the answerable subsets of these datasets.

**HotpotQA** (Yang et al., 2018) is a large-scale Wikipedia-based multi-hop QA benchmark comprising approximately 113K question-answer pairs. Unlike single-hop datasets, it requires reasoning over evidence from multiple documents. The benchmark further provides sentence-level supporting-fact annotations, enabling supervision for evidence grounding and interpretability analysis. In addition, HotpotQA includes comparison questions that require cross-document evidence extraction and comparative reasoning. The dataset therefore evaluates not only answer accuracy but also the ability to identify supporting evidence and perform multi-hop reasoning.

**2WikiMultiHopQA** (Ho et al., 2020) is a large-scale multi-hop QA benchmark consisting of 192,606 examples. It combines structured knowledge from Wikidata with unstructured text from Wikipedia abstracts, using logical rules and templates to generate diverse question types, including comparison, inference, composition, and bridge-comparison questions. The dataset also provides explicit evidence paths from the question to the answer, which support reasoning interpretability and allow evaluation of reasoning quality. Compared with earlier multi-hop benchmarks, 2WikiMultiHopQA places greater emphasis on explicit multi-step dependencies, making it a challenging benchmark for cross-step reasoning.

**MuSiQue** (Trivedi et al., 2022) is specifically proposed to mitigate shortcut reasoning in multi-hop QA. It adopts a bottom-up construction strategy, systematically composing single-hop questions from existing datasets into 2- to 4-hop queries while ensuring that later reasoning steps depend on information resolved in earlier hops. Although MuSiQue provides both answerable and unanswerable settings, we use the answerable subset for evaluation. Compared with other multi-hop benchmarks, MuSiQue places stronger emphasis on the necessity of each reasoning step, reducing the likelihood that models succeed through superficial pat-

tern matching. It therefore serves as a more rigorous benchmark for chain-structured reasoning.

Due to computational constraints, we evaluate on 500 instances randomly sampled from the validation split of each dataset.

### A.2 Evaluation Metrics

Following prior work on multi-hop question answering (Yu et al., 2024; Jin et al., 2025; Zhang et al., 2026), we adopt Exact Match (EM) and token-level F1 as our main evaluation metrics.

#### A.2.1 EM

EM assesses whether the predicted answer exactly matches the gold answer string. Let  $N$  denote the total number of evaluation instances,  $\hat{y}_i$  the predicted answer for the  $i$ -th instance, and  $y_i$  the corresponding ground-truth answer. EM is defined as:

$$\text{EM} = \frac{1}{N} \sum_{i=1}^N \mathbf{1}(\hat{y}_i = y_i), \quad (16)$$

where  $\mathbf{1}(\cdot)$  is the indicator function.

#### A.2.2 F1 Score

Token-level F1 measures the token overlap between the predicted answer and the ground-truth answer, and is defined as the harmonic mean of precision  $P$  and recall  $R$ :

$$\text{F1}(y, \hat{y}) = \frac{2PR}{P + R}, \quad (17)$$

where precision  $P$  and recall  $R$  are computed as

$$P = \frac{|\text{CommonTokens}(y, \hat{y})|}{|\text{Tokens}(\hat{y})|}, \quad (18)$$

$$R = \frac{|\text{CommonTokens}(y, \hat{y})|}{|\text{Tokens}(y)|}. \quad (19)$$

where  $\text{Tokens}(\cdot)$  returns the token sequence of an answer string, and  $\text{CommonTokens}(y, \hat{y})$  denotes the overlapping tokens between the prediction and the ground truth. Compared with EM, F1 provides a more fine-grained assessment by accounting for partial matches.

We do not adopt evaluation frameworks based on external large language model (LLM) judges. While the LLM-as-a-Judge paradigm offers flexibility for open-ended generation, its outputs can be affected by model bias, prompt sensitivity, and stochastic variation, which may reduce the reproducibility and reliability of cross-method comparisons (Chen et al., 2024a).

### A.3 Baseline Descriptions

We evaluate CloRe against representative multi-hop RAG baselines, grouped into two categories: non-RL methods and search-based RL methods.

#### Non-RL Baselines:

- (1) **NaiveRAG** (Lewis et al., 2020): retrieves documents using the original question and directly generates the final answer from the retrieved context.
- (2) **IRCoT** (Trivedi et al., 2023): interleaves reasoning and retrieval, using current reasoning to guide subsequent retrieval.
- (3) **Self-Ask** (Press et al., 2023): decomposes a question into explicit sub-questions and answers them step by step.
- (4) **GenGround** (Shi et al., 2024): adopts a two-stage framework that first generates candidate answers and then retrieves evidence to verify and refine them, thereby mitigating error propagation.
- (5) **DEC** (Ji et al., 2025): decomposes the input into a sub-question chain and performs step-wise retrieval and reasoning with query rewriting and discriminative keyword extraction.

#### Search-RL Baselines:

- (6) **Search-R1** (Jin et al., 2025): formulates multi-hop QA as an interleaved reasoning–search process, using reinforcement learning to optimize query generation, search invocation, and evidence use during reasoning.
- (7) **StepSearch** (Zheng et al., 2025): optimizes search-based reasoning with step-wise PPO and finer-grained intermediate rewards to improve multi-hop retrieval trajectories.
- (8) **ReasonRAG** (Zhang et al., 2026): adopts a process-supervised agentic RAG framework with fine-grained supervision over intermediate steps such as query generation, evidence extraction, and answer generation.
- (9) **GlobalRAG** (Luo et al., 2025): emphasizes global planning by decomposing the problem into sub-goals and coordinating retrieval, reasoning, and iterative evidence updates under reinforcement learning.

## B Implementation Details

### B.1 Runtime and Inference Setup

We use a unified runtime configuration across all experiments. Generation models are served via vLLM (Kwon et al., 2023), and the decoding temperature is set to 0.1 during inference. Unless oth-

erwise stated, all answer-generation experiments for the main QA system are conducted on a single NVIDIA A6000 GPU. Planner data construction is performed separately and is described in Appendix B.2.3.

### B.2 Question Planner Training Details

#### B.2.1 Data Construction

The question planner generates a sequence of sub-questions:

$$Y = \{(q_1, r_1), (q_2, r_2), \dots, (q_n, r_n)\}, \quad (20)$$

where  $q_i$  denotes the  $i$ -th sub-question and  $r_i$  indicates whether the step requires external retrieval. When a subsequent step depends on a prior intermediate answer, that dependency is explicitly referenced via a placeholder (e.g., #i). The training data for the question planner are constructed from HotpotQA, 2WikiMultiHopQA, and MuSiQue.

**SFT Data Construction.** All planner training data are derived exclusively from the official training splits of HotpotQA, 2WikiMultiHopQA, and MuSiQue; no validation examples are used at any stage. For each training question, we employ Qwen3-32B to generate a sub-question sequence with explicit dependency references and retrieval indicators. Under a unified output-format constraint, we filter malformed or invalid decompositions and retain only structurally valid and executable instances. This yields 5,380, 1,454, and 1,148 SFT examples from HotpotQA, 2WikiMultiHopQA, and MuSiQue, respectively, for a total of 7,982 instances.

**DPO Data Construction.** Using the same training-only splits, we construct multiple candidate decompositions for each question and execute them under a unified decompose–retrieve–answer pipeline. Candidates are obtained from greedy SFT outputs, sampled SFT outputs, and teacher-generated decompositions from Qwen3-32B. We discard structurally invalid candidates, including those with unused intermediate variables, illegal forward references, or format violations, and rank the remaining candidates by final answer quality, breaking ties in favor of shorter decompositions. This yields 816, 113, and 399 preference pairs from HotpotQA, 2WikiMultiHopQA, and MuSiQue, respectively, totaling 1,328 pairs. The SFT and DPO data are pooled across the three datasets to train a single planner for question decomposition, dependency prediction, and retrieval-indicator prediction.

| Method      | HotpotQA                      |                              | 2WikiMultiHopQA               |                               | MuSiQue                       |                               |
|-------------|-------------------------------|------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|
|             | EM                            | F1                           | EM                            | F1                            | EM                            | F1                            |
| <b>Ours</b> | <b>57.20</b>                  | <b>64.26</b>                 | <b>63.70</b>                  | <b>71.03</b>                  | <b>34.80</b>                  | <b>41.57</b>                  |
| w/o SFT     | 51.30 ( $\downarrow 10.3\%$ ) | 59.53 ( $\downarrow 7.4\%$ ) | 54.20 ( $\downarrow 14.9\%$ ) | 62.34 ( $\downarrow 12.2\%$ ) | 26.20 ( $\downarrow 24.7\%$ ) | 33.48 ( $\downarrow 19.5\%$ ) |
| w/o DPO     | 52.40 ( $\downarrow 8.4\%$ )  | 59.91 ( $\downarrow 6.8\%$ ) | 63.60 ( $\downarrow 0.2\%$ )  | 70.91 ( $\downarrow 0.2\%$ )  | 33.20 ( $\downarrow 4.6\%$ )  | 40.06 ( $\downarrow 3.6\%$ )  |

Table 9: Planner training ablation on Qwen2.5-7B-Instruct. *w/o SFT* removes supervised fine-tuning, while *w/o DPO* removes preference optimization. Numbers in parentheses denote the relative drop compared with full model.

## B.2.2 Training Objectives

The SFT stage minimizes the standard autoregressive negative log-likelihood objective:

$$\mathcal{L}_{\text{SFT}} = - \sum_{t=1}^T \log p_{\theta}(y_t^* | x, y_{<t}^*), \quad (21)$$

where  $x$  denotes the input question and  $y^*$  the target decomposition sequence.

The DPO stage then optimizes the model using the standard preference objective:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x, y^+, y^-)} \left[ \log \sigma \left( \beta \left( \log \frac{\pi_{\theta}(y^+ | x)}{\pi_{\text{ref}}(y^+ | x)} - \log \frac{\pi_{\theta}(y^- | x)}{\pi_{\text{ref}}(y^- | x)} \right) \right) \right], \quad (22)$$

where  $\pi_{\text{ref}}$  denotes the reference policy derived from the SFT stage, and  $\beta$  is the temperature parameter. This formulation optimizes the policy on preference pairs without requiring an auxiliary reward model.

## B.2.3 Implementation and Hyperparameters

We use Qwen3-32B (Yang et al., 2025) as the teacher model to generate supervision for question decomposition, and Qwen3-4B-Instruct (Yang et al., 2025) as the backbone model of the question planner. Both training stages are implemented with LLaMA-Factory (Zheng et al., 2024) and utilize LoRA (Hu et al., 2022) for parameter-efficient fine-tuning. The learning rates for SFT and DPO are set to  $5 \times 10^{-5}$  and  $5 \times 10^{-6}$ , respectively, and both stages are trained for 5 epochs. All training runs are conducted on a single NVIDIA A6000 GPU. Candidate decompositions from the teacher model are generated on a server with four NVIDIA A40 GPUs, each with 46 GB of memory.

## B.2.4 Planner Training Ablation

To analyze the impact of the question planner’s two-stage training strategy on overall system performance, we conduct ablation studies on the Qwen2.5-7B-Instruct backbone. The results are

## Algorithm 1 Closed-Loop Inference of CloRe

**Require:** Question  $x$ ; planner  $QP$ ; global budget  $B_{\text{max}}$ ; stage schedule  $\{p_j\}_{j=0}^K$ ; local rewrite budget  $R_{\text{max}}$

**Ensure:** Final answer  $\hat{y}$

```

1:  $Y = \{(q_t, r_t)\}_{t=1}^T \leftarrow QP(x)$ 
2:  $H_0 \leftarrow \emptyset$ ,  $B_0 \leftarrow \emptyset$ ,  $t \leftarrow 1$ ,  $b \leftarrow 0$ 
3: while  $t \leq T$  and  $b < B_{\text{max}}$  do
4:   Construct  $H_{t-1}^{\text{dep}}$  and  $B_{t-1}^{\text{dep}}$  from  $(H_{t-1}, B_{t-1})$ 
5:    $\tilde{q}_t \leftarrow W(q_t, H_{t-1}^{\text{dep}})$ 
6:    $\rho_t \leftarrow 0$ , committed  $\leftarrow$  false
7:   if  $r_t = 0$  then
8:      $\hat{a}_t \leftarrow G(\tilde{q}_t, H_{t-1}^{\text{dep}})$ 
9:      $H_t \leftarrow H_{t-1} \cup \{(t, q_t, \tilde{q}_t, \hat{a}_t)\}$ 
10:     $B_t \leftarrow B_{t-1}$ 
11:     $t \leftarrow t + 1$ 
12:    continue
13:   end if
14:   while not committed and  $b < B_{\text{max}}$  do
15:     for  $j = 0$  to  $K$  do
16:       Retrieve candidates for  $\tilde{q}_t$  and construct stage
       evidence  $E_t^{(j)}$ 
17:        $\hat{a}_t^{(j)} \leftarrow G(\tilde{q}_t, E_t^{(j)})$ 
18:        $o_t^{(j)} \leftarrow \mathcal{A}(q_t, \tilde{q}_t, \hat{a}_t^{(j)}, E_t^{(j)}, H_{t-1}^{\text{dep}})$ 
19:       if  $o_t^{(j)} = \text{accept}$  then
20:          $\hat{a}_t \leftarrow \hat{a}_t^{(j)}$ 
21:         committed  $\leftarrow$  true
22:         break
23:       else if  $o_t^{(j)} = \text{revise}$  then
24:          $\hat{a}_t \leftarrow \text{Revise}(o_t^{(j)})$ 
25:         committed  $\leftarrow$  true
26:         break
27:       end if
28:     end for
29:     if committed then
30:        $H_t \leftarrow H_{t-1} \cup \{(t, q_t, \tilde{q}_t, \hat{a}_t)\}$ 
31:        $B_t \leftarrow B_{t-1}$ 
32:        $t \leftarrow t + 1$ 
33:     else if  $\rho_t < R_{\text{max}}$  then
34:        $\tilde{q}_t \leftarrow W_{\text{retry}}(q_t, H_{t-1}^{\text{dep}}, B_{t-1}^{\text{dep}}, f_t)$ 
35:        $\rho_t \leftarrow \rho_t + 1$ 
36:     else
37:       Select an explicit predecessor hop  $k_t^* \in \mathcal{D}_t$ 
38:       Retrieve the accepted answer  $\hat{a}_{k_t^*}$  from  $H_{t-1}$ 
39:        $B_t \leftarrow B_{t-1} \cup \{(k_t^*, q_{k_t^*}, \tilde{q}_{k_t^*}, \hat{a}_{k_t^*})\}$ 
40:        $H_t \leftarrow \text{Rollback}(H_{t-1}, k_t^*)$ 
41:        $t \leftarrow k_t^*$ 
42:     end if
43:      $b \leftarrow b + 1$ 
44:   end while
45: end while
46: Let  $H_{\tau}$  denote the committed history at termination
47:  $\hat{y} \leftarrow G_{\text{agg}}(x, H_{\tau})$ 
48: return  $\hat{y}$ 

```

reported in Table 9, from which we draw two observations.

**(1) SFT is critical for stable decomposition.** Removing the SFT stage (w/o SFT) causes substantial performance drops across all three datasets, with relative decreases of 10.3%–24.7% in EM and 7.4%–19.5% in F1. The degradation is especially pronounced on 2WikiMultiHopQA and MuSiQue, suggesting that without supervised decomposition signals, the planner has difficulty learning stable sub-question structures, explicit dependency references, and reliable retrieval indicators.

**(2) DPO further improves decomposition quality on top of SFT.** Removing the DPO stage (w/o DPO) also reduces performance relative to the full model, although the drop is much smaller than that of w/o SFT. Specifically, performance decreases by 8.4% in EM and 6.8% in F1 on HotpotQA, and by 4.6% in EM and 3.6% in F1 on MuSiQue, while only marginal declines are observed on 2WikiMultiHopQA (0.2% in EM and 0.2% in F1). This pattern suggests that SFT already provides strong structured decomposition ability, whereas DPO mainly refines preference ranking among candidate decompositions, favoring outputs that are more compact and better aligned with downstream reasoning.

## C Full Algorithm and Additional Method Details

Algorithm 1 outlines the full closed-loop inference procedure of CloRe. Initialized with the planner-generated sub-question sequence, the framework performs dependency-aware forward reasoning with stage-wise evidence acquisition and step-level review. When local remediation is exhausted, CloRe triggers backward correction along explicit dependencies, rolls back the affected history, and resumes forward reasoning from the updated state.

**Global Inference Budget.** To control the overall inference cost, CloRe adopts a fixed global inference budget  $B_{\max} = 8$  in all experiments. The global counter accumulates the total number of executed sub-question solving attempts and is never reset during inference, including after backtracking. Note that the counters reset after backtracking are only local counters, such as the evidence-expansion stage index and retry-rewriting count. For example, solving a sequence of sub-questions  $q_1 \rightarrow q_2 \rightarrow q_3$

consumes three budget units. If  $q_3$  triggers backtracking to  $q_2$ , re-executing  $q_2$  and  $q_3$  consumes two additional units rather than restarting the budget counter. The average consumed budgets are 2.35, 3.21, and 3.77 on HotpotQA, 2WikiMultiHopQA, and MuSiQue, respectively, remaining well below the predefined budget limit.

**Retry Rewriting.** The retrieval query at hop  $t$  is not rewritten by default. Retry rewriting is triggered only when all evidence stages are exhausted, the reviewer returns a reject decision for the current candidate, and the current hop still satisfies  $\rho_t < R_{\max}$ . In that case, the framework invokes a failure-driven rewriter  $W_{\text{retry}}(\cdot)$  to reformulate the query, enriching underspecified constraints or improving retrieval alignment.

**Stage-0 Evidence Compression.** Stage 0 prioritizes compact evidence over broad document coverage, since long contexts can remain a bottleneck for small-to-mid-sized LLMs (Shi et al., 2024). Given the retrieval query  $\tilde{q}_t$  at hop  $t$ , CloRe extracts salient query units using the spaCy toolkit, including named entities, numeric mentions, comparative or superlative expressions, and consecutive proper-noun spans, and normalizes them into lowercase strings. These query units are then used to filter the candidate documents selected under the initial top- $p_0$  threshold, retaining only those that overlap with the extracted units.

Formally, let  $\mathcal{K}(\tilde{q}_t)$  denote the set of query units extracted from  $\tilde{q}_t$ , and let  $D_t^{(0)}$  denote the candidate documents selected under the initial top- $p_0$  threshold. The Stage-0 compression operator  $\Psi(\cdot)$  returns the subset:

$$\Psi(D_t^{(0)}, \tilde{q}_t) = \{d \in D_t^{(0)} \mid \text{Match}(d, \mathcal{K}(\tilde{q}_t)) = 1\}. \quad (23)$$

where  $\text{Match}(\cdot)$  indicates whether a document contains at least one extracted query unit.

## D Additional Efficiency Analysis

Tables 10–12 report additional inference-efficiency statistics, including model calls, retrieval calls, generated tokens, and total tokens. The comparison includes representative iterative and RL-search methods. Latency is reported only for non-RL methods under the same backbone, hardware, concurrency, and serving setup, since the RL agents use different servers and serving stacks.

**(1) Model and retrieval calls.** On HotpotQA and 2WikiMultiHopQA, CloRe is comparable to DEC and more efficient than GenGround, indicating that

| Method     | EM          | F1           | Latency (s) | LLM Calls / Gen. Rounds | Ret. Calls | Gen. Tokens | Total Tokens |
|------------|-------------|--------------|-------------|-------------------------|------------|-------------|--------------|
| GenGround  | 46.1        | 55.59        | 32.23       | 12.60                   | 4.96       | 776         | 7,289        |
| DEC        | 49.0        | 57.85        | 17.95       | 9.41                    | 2.47       | 322         | 1,389        |
| Search-R1  | 55.3        | 63.10        | –           | 3.24                    | 2.25       | 226         | 2,215        |
| StepSearch | 54.5        | 61.23        | –           | 2.83                    | 1.83       | 185         | 1,759        |
| CloRe      | <b>57.2</b> | <b>63.83</b> | 10.23       | 8.88                    | 2.36       | 198         | 5,035        |

Table 10: End-to-end effectiveness and inference-cost statistics on HotpotQA.

| Method     | EM          | F1           | Latency (s) | LLM Calls / Gen. Rounds | Ret. Calls | Gen. Tokens | Total Tokens |
|------------|-------------|--------------|-------------|-------------------------|------------|-------------|--------------|
| GenGround  | 45.6        | 58.25        | 31.27       | 13.11                   | 4.96       | 764         | 7,723        |
| DEC        | 55.6        | 64.04        | 20.85       | 9.65                    | 2.55       | 331         | 3,909        |
| Search-R1  | 53.5        | 61.13        | –           | 3.65                    | 2.65       | 255         | 3,331        |
| StepSearch | 49.7        | 56.98        | –           | 3.28                    | 2.28       | 190         | 2,592        |
| CloRe      | <b>63.8</b> | <b>71.80</b> | 14.26       | 10.86                   | 3.05       | 232         | 7,123        |

Table 11: End-to-end effectiveness and inference-cost statistics on 2WikiMultiHopQA.

review and correction do not consistently inflate execution. On MuSiQue, more frequent evidence expansion, rewriting, and backtracking increase both call types. Search-R1 and StepSearch use fewer generation rounds, but their different interaction structures make direct comparison with non-RL LLM calls imperfect.

**(2) Token usage.** CloRe maintains moderate generated-token usage, comparable to RL-search methods. Its higher total cost mainly comes from repeatedly processing evidence and historical states during generation, review, and correction. Evidence selection keeps total usage below GenGround on HotpotQA and 2WikiMultiHopQA, though it remains above DEC, Search-R1, and StepSearch.

## E Additional Analysis of Execution Mechanisms and Strategy Choices

### E.1 Retrieval Probability Distribution Analysis

In the proposed evidence-evolution mechanism, CloRe does not consume all retrieved documents at once; instead, it starts from a compact evidence set and expands the scope only when the current context is insufficient. To better motivate this design, we analyze the relevance distribution of retrieval-required sub-questions to inform the choice of the Stage-0 starting point  $p_0$  and the subsequent expansion schedule.

For each retrieval-required sub-question, let

$$\mathbf{s} = (s_1, s_2, \dots, s_K) \quad (24)$$

denote the normalized relevance distribution over

the top- $K$  retrieved documents, where  $s_i$  is the probability assigned to the  $i$ -th document. We further define the cumulative probability mass function as:

$$C(k) = \sum_{i=1}^k s_i, \quad (25)$$

which measures the total probability mass covered by the top- $k$  documents.

We compute this distribution over the top-10 retrieved documents for retrieval-required sub-questions from HotpotQA, 2WikiMultiHopQA, and MuSiQue, covering 2,175, 2,741, and 6,304 sub-questions, respectively. Figure 5 shows the average cumulative probability mass curves across the three datasets. All three curves rise steeply in the top ranks, indicating that highly relevant evidence is concentrated in a small number of top-ranked documents, although the degree of concentration differs across datasets. HotpotQA and 2WikiMultiHopQA exhibit sharper early growth, whereas MuSiQue shows a flatter curve, suggesting that its relevant evidence is more dispersed and that broader evidence coverage may be more important.

Specifically, the average top-1 probability mass reaches 82.86%, 84.32%, and 72.10% for HotpotQA, 2WikiMultiHopQA, and MuSiQue, respectively. By top-3, the cumulative mass increases to 95.02%, 96.67%, and 89.17%, while MuSiQue requires a broader prefix to reach the same coverage achieved by the other two datasets. The specific parameterization of the Stage-0 starting point and its efficiency implications are detailed in Appendix E.2.

| Method     | EM          | F1           | Latency (s) | LLM Calls / Gen. Rounds | Ret. Calls | Gen. Tokens | Total Tokens |
|------------|-------------|--------------|-------------|-------------------------|------------|-------------|--------------|
| GenGround  | 19.1        | 29.24        | 24.37       | 15.20                   | 4.92       | 863         | 8,705        |
| DEC        | 24.1        | 31.49        | 13.54       | 7.06                    | 2.53       | 353         | 3,610        |
| Search-R1  | 28.2        | 35.98        | —           | 3.07                    | 2.07       | 215         | 2,664        |
| StepSearch | 29.6        | 36.27        | —           | 3.09                    | 2.09       | 189         | 2,278        |
| CloRe      | <b>31.4</b> | <b>37.96</b> | 15.36       | 25.63                   | 6.00       | 438         | 22,915       |

Table 12: End-to-end effectiveness and inference-cost statistics on MuSiQue.

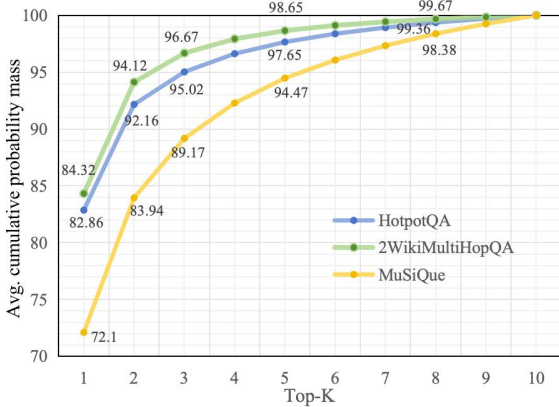


Figure 5: Average cumulative retrieval probability mass over Top- $k$  documents on retrieval-required sub-questions. All three datasets show strong head concentration, while MuSiQue exhibits a flatter curve, indicating more dispersed relevant evidence.

## E.2 Choice of Initial Point $p_0$

In the evidence-evolution framework, Stage 0 is designed to construct an initial evidence set that is as compact as possible while still supporting preliminary answer generation. The choice of  $p_0$  therefore controls a trade-off between performance and efficiency: a smaller  $p_0$  more often triggers later-stage expansion because the initial context may provide less support, whereas a larger  $p_0$  increases the Stage-0 context size and may introduce more irrelevant evidence. We therefore evaluate multiple configurations of  $p_0$ .

Unlike the dynamic evidence-evolution process used in the main system, the calibration of  $p_0$  follows a controlled two-stage setup. Stage 0 first extracts an initial evidence set under a given  $p_0$ . If this context is insufficient, the framework directly expands to a fixed, high-recall, non-compressed evidence pool. We set this second-stage threshold to  $p = 0.99$  rather than  $p = 1.0$  for two reasons. First, isolating the effect of  $p_0$  requires the fallback context size to remain fixed. Second, as shown in Appendix E.1, the average cumulative probability mass at top-8 already approaches 0.99

across all three datasets, making  $p = 0.99$  a near-complete pool while avoiding the larger context size and potentially weaker tail evidence associated with  $p = 1.0$ .

We utilize EM, F1, and the average document count per retrieval-required sub-question as evaluation metrics, where EM and F1 measure answer quality and the document count reflects the cost of the Stage-0 context. Figure 6 plots the resulting performance–cost trade-offs across different values of  $p_0$ . The preferred value of  $p_0$  varies slightly across datasets, but  $p_0 = 0.3$  remains consistently competitive and provides the best unified choice overall. HotpotQA and 2WikiMultiHopQA reach peak EM/F1 at  $p_0 = 0.3$ , whereas MuSiQue remains competitive over a broader range ( $0.3 \leq p_0 \leq 0.5$ ) but incurs noticeably larger document counts as  $p_0$  increases. At  $p_0 = 0.3$ , the average Stage-0 document count is 1.29, 1.08, and 2.20 for HotpotQA, 2WikiMultiHopQA, and MuSiQue, respectively. We therefore use  $p_0 = 0.3$  as the unified starting threshold in the main system.

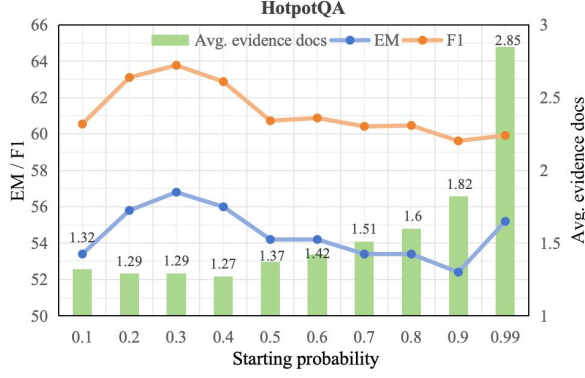
## E.3 Design of the Evidence-Evolution Schedule

Following the calibration of the unified starting point  $p_0 = 0.3$  in Appendix E.2, we define the complete evidence-evolution schedule as

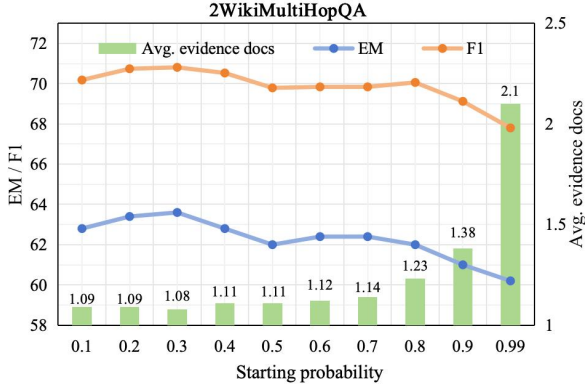
$$p = [0.3, 0.9, 0.99, 1.0].$$

Stage 0 establishes a compact initial evidence set; Stage 1 expands the context to include more supporting information; Stage 2 provides a broader fallback evidence pool; and Stage 3 uses the full top-10 retrieved results as the final upper bound. This schedule allows the system to begin from a compact initial context and expand the retrieval scope when the current evidence does not support a committable answer.

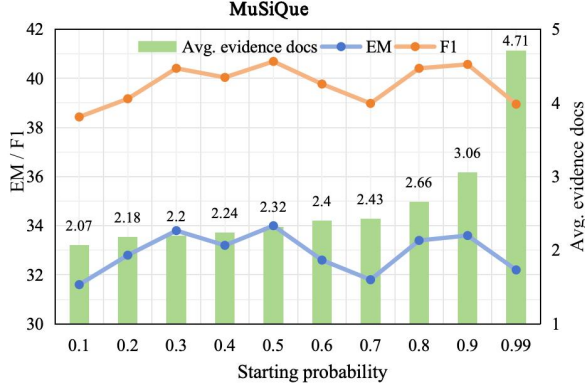
The corresponding evidence scales are summarized in Table 13. The controlled two-stage setup used in Appendix E.2 is only for calibrating the initial point  $p_0$ , whereas the schedule



(a) Effectiveness–cost trade-off under different starting probabilities  $p_0$  on HotpotQA.



(b) Effectiveness–cost trade-off under different starting probabilities  $p_0$  on 2WikiMultiHopQA.



(c) Effectiveness–cost trade-off under different starting probabilities  $p_0$  on MuSiQue.

Figure 6: Effectiveness–cost trade-off under different starting probabilities  $p_0$ .

[0.3, 0.9, 0.99, 1.0] is the full evidence-evolution sequence used in the main experiments.

#### E.4 Stage Triggering and Efficiency under Evidence-Only Evolution

This section examines the triggering behavior and execution overhead of evidence evolution under a controlled setting where query rewriting and local backtracking are disabled. The stage-trigger statis-

| Stage | Probability( $p$ ) | Document Count |
|-------|--------------------|----------------|
| 0     | 0.30               | 1–2            |
| 1     | 0.90               | 2–3            |
| 2     | 0.99               | 4–8            |
| 3     | 1.00               | 10             |

Table 13: Probability schedule of evidence evolution and the corresponding document constraints at each stage.

| Dataset  | Review Cases | Fleiss’ $\kappa$ | Disagree Rate |
|----------|--------------|------------------|---------------|
| HotpotQA | 1,068        | 0.9855           | 0.0075        |
| 2Wiki    | 1,306        | 0.9901           | 0.0023        |
| MuSiQue  | 1,506        | 0.9791           | 0.0133        |

Table 14: Stability of reviewer decisions over three repeated runs per review case.

tics reported therefore characterize evidence-only evolution and are not directly comparable to the execution statistics of the full system.

Under the unified schedule [0.3, 0.9, 0.99, 1.0], we evaluate stage triggering and average execution cost for retrieval-required sub-questions. The Stage- $i$  trigger ratio denotes the proportion of retrieval-required sub-questions that enter Stage  $i$ ; Avg. executed stages represents the average number of executed stages per sub-question; and Avg. evidence docs denotes the average number of evidence documents used per sub-question.

Figure 7 shows that Stage 0 is triggered for all retrieval-required sub-questions, while trigger ratios decline rapidly at deeper stages. For HotpotQA, the trigger ratios of Stage 1, Stage 2, and Stage 3 are 10.54%, 5.49%, and 2.85%, respectively; for 2WikiMultiHopQA, they are 6.35%, 2.61%, and 1.06%; and for MuSiQue, they are 11.37%, 5.23%, and 2.37%. This distribution indicates that deeper expansion is required only for a limited subset of retrieval-required sub-questions.

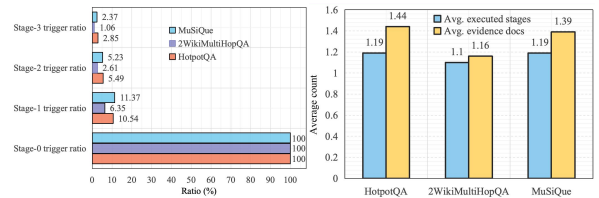


Figure 7: Stage triggering and execution efficiency under the evidence-only evolution setting. Left: Stage trigger ratios on retrieval-required sub-questions. Right: average executed stages and average evidence documents per retrieval-required sub-question.

| Dataset  | Cases | Accuracy | Macro-F1 | Cohen’s $\kappa$ | Accept Recall | Revise Recall | Reject Recall |
|----------|-------|----------|----------|------------------|---------------|---------------|---------------|
| HotpotQA | 748   | 86.23    | 76.00    | 0.673            | 94.12         | 70.37         | 59.38         |
| 2Wiki    | 829   | 93.24    | 71.86    | 0.616            | 95.90         | 69.23         | 63.04         |
| MuSiQue  | 908   | 80.07    | 75.35    | 0.650            | 90.70         | 68.24         | 64.41         |

Table 15: Evaluation of the default 7B reviewer against human annotations. Accuracy and recall are reported in percentages.

Average execution cost remains low across all three datasets. The values of Avg. executed stages are 1.19, 1.10, and 1.19 for HotpotQA, 2WikiMultiHopQA, and MuSiQue, respectively, all close to 1.0 and well below the maximum depth of 4. The corresponding values of Avg. evidence docs are 1.44, 1.16, and 1.39. Together with the effectiveness results in the main text, these statistics suggest that evidence evolution can improve performance without requiring most instances to traverse the full expansion schedule.

### E.5 Reviewer Decision Stability

Table 14 reports the stability of reviewer decisions over three repeated runs for each review case. The reviewer uses the same temperature of 0.1 as in the main experiments.

### E.6 Detailed Reviewer Correctness Analysis

Three annotators independently labeled each review case, with disagreements resolved through adjudication. A case is labeled accept if the candidate answer is correct, revise if it is incorrect but recoverable from the current context, and reject if the current context is insufficient for correction.

The gold-label distributions of accept/revise/reject are 72.73/14.44/12.83%, 91.31/3.14/5.55%, and 56.83/18.72/24.45% on HotpotQA, 2WikiMultiHopQA, and MuSiQue, respectively. Given this class imbalance, we report macro-F1, Cohen’s  $\kappa$ , and class-wise recall in addition to accuracy. As shown in Table 15, the reviewer achieves macro-F1 scores of 71.86–76.00 and Cohen’s  $\kappa$  values of 0.616–0.673, while retaining meaningful recall on revise and reject cases.

We further evaluate the reviewer’s two functional roles in CloRe. Gold revise and reject cases are treated as incorrect intermediate answers, and an error is successfully detected when the reviewer predicts either label. Among detected errors, recovery-selection accuracy measures whether the predicted revise/reject action matches the human label. As shown in Table 16, the reviewer detects 75.00–

| Dataset  | Error-Detection Recall | Recovery-Selection Acc. |
|----------|------------------------|-------------------------|
| HotpotQA | 156/204 (76.47%)       | 133/156 (85.26%)        |
| 2Wiki    | 54/72 (75.00%)         | 47/54 (87.04%)          |
| MuSiQue  | 306/392 (78.06%)       | 259/306 (84.64%)        |

Table 16: Reviewer correctness in error detection and recovery selection. Recovery-selection accuracy is conditioned on detected errors.

| Dataset  | Gold   | Pred. Accept | Pred. Revise | Pred. Reject |
|----------|--------|--------------|--------------|--------------|
| HotpotQA | Accept | 512          | 19           | 13           |
|          | Revise | 24           | 76           | 8            |
|          | Reject | 24           | 15           | 57           |
| 2Wiki    | Accept | 726          | 14           | 17           |
|          | Revise | 6            | 18           | 2            |
|          | Reject | 12           | 5            | 29           |
| MuSiQue  | Accept | 468          | 28           | 20           |
|          | Revise | 38           | 116          | 16           |
|          | Reject | 48           | 31           | 143          |

Table 17: Confusion matrices of the default 7B reviewer against human annotations. Rows denote human labels and columns denote reviewer predictions.

78.06% of incorrect intermediate answers and selects the correct recovery action for 84.64–87.04% of detected errors.

Table 17 reports the complete confusion matrices. Most errors involve confusing an erroneous state with accept, while confusions between revise and reject are relatively less frequent.

## F Case Study

To better understand the dynamic behavior of CloRe along multi-hop reasoning chains, we present several representative qualitative case studies. Figure 8 shows a case in which the initial decomposition and shallow evidence are sufficient for direct resolution at Stage 0. As depicted in Figure 9, a local inconsistency between an intermediate answer and its supporting evidence is corrected by the reviewer without triggering deeper evidence expansion. Figures 10 and 11 present cases where shallow evidence is insufficient, leading the sys-

tem to progressively expand the evidence scope and, when necessary, apply query rewriting to improve retrieval alignment. Figure 12 illustrates a backtracking case in which persistent downstream failure acts as a reverse signal for a predecessor state: the answer *Rome* is first produced for q2 and passes local review, but later downstream failure triggers dependency-aware backtracking and correction.

## **G Prompts**

This section presents the prompt templates used in our experiments. These templates cover the main modules of CloRe, including candidate decomposition generation, conditional rewriting, failure-driven retry rewriting, step-level review, backward correction, and final answer aggregation. Note that the examples given in the prompts were not part of the dataset used for testing.

| Case study for CloRe of straightforward Stage 0 reasoning                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Question:</b> Macclesfield Town F.C. plays in the league sponsored by whom?</p> <p><b>Ground Truth:</b> ["Vanarama"]</p> <hr/> <p><b>Problem Planning</b></p> <p>q1: Which league does Macclesfield Town F.C. play in? [Retrieve]</p> <p>q2: Who sponsors the league #1? [Retrieve]</p> <hr/> <p><b>Reasoning Steps</b></p> <p><b>Step 1</b></p> <p>q1: Which league does Macclesfield Town F.C. play in?</p> <p><b>Key Units:</b> ["Macclesfield Town F.C.", "league"]</p> <p><b>Stage:</b> Stage 0</p> <p><b>Selected Evidence:</b> Doc 1 (Title: <u>Macclesfield F.C.</u>) Macclesfield Town Football Club is a professional association football club based in Macclesfield, Cheshire, England. The club currently plays in the <u>National League</u>, the fifth tier of English football.</p> <p>a1: <b>National League</b></p> <p><b>Reviewer:</b> Accept</p> <hr/> <p><b>Step 2</b></p> <p>Rewritten q2: Who sponsors the National League?</p> <p><b>Key Units:</b> ["National League"]</p> <p><b>Stage:</b> Stage 0</p> <p><b>Selected Evidence:</b> Doc 1 (Title: <u>National League (English football)</u>) The <u>National League</u> is an association football league in England consisting of three divisions, the <u>National League</u>, <u>National League North</u> and <u>National League South</u>. It was called the Alliance Premier League from 1979 until 1986. Between 1986 and 2015, the league was known as the Football Conference. As part of a sponsorship deal with car leasing company Vanarama, the league is known as the <u>Vanarama National League</u>.</p> <p>a2: <b>Vanarama</b></p> <p><b>Reviewer:</b> Accept</p> <hr/> <p><b>Final Answer:</b> <b>Vanarama</b></p> |

Figure 8: CloRe first decomposes the original question into a dependency-aware sub-question chain, followed by sequential sub-question rewriting, retrieval, answering, and review. Since all sub-questions obtain evidence sufficient to support the staged answers at Stage 0, no further evidence expansion is triggered. The final answer is produced by aggregating the original question, completed sub-questions, and corresponding sub-answers. Blue text indicates a supporting document containing a staged answer, and underlined text indicates key units in the document. Orange text indicates sub-questions used in final aggregation, and green text indicates the corresponding sub-answers.

### Case study for CloRe of reviewer-guided intermediate-answer correction

**Question:** Old Bushmills Distillery was in the process of transitioning from the ownership of the company that had a headquarters in what English city?

**Ground Truth:** ["London"]

#### Problem Planning

**q1:** Which company was Old Bushmills Distillery transitioning from ownership by? [Retrieve]

**q2:** In what English city is the headquarters of #1 located? [Retrieve]

#### Reasoning Steps

##### Step 1

**q1:** Which company was Old Bushmills Distillery transitioning from ownership by?

**Key Units:** ["Old Bushmills Distillery", "ownership"]

**Stage:** Stage 0

**Selected Evidence:** Doc 1 (Title: Old Bushmills Distillery) The Old Bushmills Distillery is a distillery in Bushmills, County Antrim, Northern Ireland...of December 2014, it was in the process of transitioning from ownership by Diageo plc to Jose Cuervo. All of the whiskey bottled under the Bushmills whiskey brand is produced at the Bushmills Distillery and uses water drawn from Saint Columb's Rill, which is a tributary of the River Bush. ...

**a1:** Jose Cuervo

**Reviewer:** Revise (**a1:** Diageo plc)

##### Step 2

**Rewritten q2:** In what English city is the headquarters of Diageo plc located?

**Key Units:** ["English", "Diageo", "plc"]

**Stage:** Stage 0

**Selected Evidence:** Doc 1 (Title: Diageo) Diageo plc is a British multinational alcoholic beverages company, with its headquarters in London, England. It was the world's largest distiller ..."

**a2:** Vanarama

**Reviewer:** Accept

**Final Answer:** London

Figure 9: After retrieving evidence in forward reasoning, CloRe first generates an intermediate answer. When this intermediate answer is inconsistent with the supporting evidence, the reviewer revises it based on the current evidence and passes the corrected result to subsequent sub-questions, so that later reasoning proceeds from a more reliable intermediate state. Purple text indicates failed sub-questions or intermediate answers.

| Case study for CloRe of stage-wise evidence expansion                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Question:</b> Who is the employer of the physicist who Mach number is named after?<br><b>Ground Truth:</b> ["Charles University"]                                                                                                                                                                                                                                                                                                          |
| <b>Problem Planning</b><br><b>q1:</b> After which physicist is the Mach number named? [Retrieve]<br><b>q2:</b> Who is the employer of #1? [Retrieve]                                                                                                                                                                                                                                                                                          |
| <b>Reasoning Steps</b><br><b>Step 1</b><br><b>q1: After which physicist is the Mach number named?</b><br><b>Key Units:</b> ["Mach"]<br><b>Stage:</b> Stage 0<br><b>Selected Evidence:</b> Doc 1 (Title: <u>Mach</u> number) <u>The Mach number is named after Austrian physicist and philosopher Ernst Mach</u> , and is a designation proposed by aeronautical engineer Jakob Ackeret...<br><b>a1: Ernst Mach</b><br><b>Reviewer:</b> Accept |
| <b>Step 2</b><br><b>Rewritten q2: Who is the employer of Ernst Mach?</b><br><b>Key Units:</b> ["Ernst Mach"]<br><b>Stage:</b> Stage 0<br><b>Selected Evidence:</b> Doc 1 (Title: Mach number) The Mach number is named after Austrian physicist and philosopher Ernst Mach, and is a designation proposed by aeronautical engineer Jakob Ackeret...<br><b>a2: Unknown</b><br><b>Reviewer:</b> Reject                                          |
| <b>Stage:</b> Stage 1<br><b>Selected Evidence:</b> Doc 1... Doc 2 (Title: Mach's principle) In theoretical physics...Doc 3 (Title: Friedrich Hayek) At the University of Vienna, Hayek earned doctorates in law and political science in 1921 and 1923 respectively...<br><b>a2: Unknown</b><br><b>Reviewer:</b> Reject                                                                                                                       |
| <b>Stage:</b> Stage 2<br><b>Selected Evidence:</b> Doc 1 ... Doc 2...Doc 3...Doc 4 (Title: Vinko Dvořák) <u>He studied mathematics and physics at the Charles University in Prague</u> , and after graduating he became an assistant to professor Ernst Mach... Doc 5 (Title: Mach bands) Mach bands is an optical illusion named after the physicist Ernst Mach...<br><b>a2: Charles University</b><br><b>Reviewer:</b> Accept               |
| <b>Final Answer:</b> <b>Charles University</b>                                                                                                                                                                                                                                                                                                                                                                                                |

Figure 10: CloRe first decomposes the original question into a dependency-aware sub-question chain and then solves the sub-questions sequentially. For the first sub-question, the system retrieves evidence containing the staged answer at Stage 0 and passes review immediately. For the second sub-question, the evidence at Stage 0 and Stage 1 does not provide sufficient support for a correct answer, so the system progressively expands the evidence scope until Stage 2, where a supporting document containing the critical information is introduced, leading to the correct sub-answer and a successful review. Gray text indicates a question-independent document.

## Case study for CloRe of recovery through evidence expansion and query rewriting

**Question:** Who owns the record label of the Another Page performer?

**Ground Truth:** ["Warner Music Group", "Warner Music"]

### Problem Planning

**q1:** Which record label does the performer of Another Page belong to? [Retrieve]

**q2:** Who is the owner of the record label #1? [Retrieve]

### Reasoning Steps

#### Step 1

**q1:** Which record label does the performer of Another Page belong to?

**Key Units:** ["Another Page", "record"]

**Stage:** Stage 0

**Selected Evidence:** Doc 1 (Title: Another Page) Another Page is Christopher Cross' second studio album, recorded in 1982 and released in early 1983...

**a1:** Unknown

**Reviewer:** Reject

**Stage:** Stage 1

**Selected Evidence:** Doc 1...Doc 2(Title: Christopher Cross) "All Right" is a song written and recorded by...

**a1:** Unknown

**Reviewer:** Reject

**Stage:** Stage 2

**Selected Evidence:** Doc 1...Doc 2...Doc 3(Title: Apple Records) Apple Records is a record label founded by...

**a1:** Unknown

**Reviewer:** Reject

**Stage:** Stage 3

**Selected Evidence:** Doc 1...Doc 2...Doc 10(Title: Roxy Recordings) King Island Roxystars Recordings AB...

**a1:** Unknown

**Reviewer:** Reject

**Rewritten q1:** Who owns the record label associated with the performer of Another Page?

**Key Units:** ["Another Page", "record"]

**Stage:** Stage 1

**Selected Evidence:** Doc 1 (Title: Another Page) Another Page is Christopher Cross' second studio album, recorded in 1982 and released in early 1983...

**a1:** Unknown

**Reviewer:** Reject

**Stage:** Stage 1

**Selected Evidence:** Doc 1...Doc 2 (Title: Back of My Mind (Christopher Cross album) Back of My Mind is the fourth studio album by singer Christopher Cross, released in 1988 through Warner Bros. Records...

**a1:** Warner Bros. Records

**Reviewer:** Accept

#### Step 2

**Rewritten q2:** Who is the owner of the record label Warner Bros. Records?

**Key Units:** ["Warner Bros. Records"]

**Stage:** Stage 0

**Selected Evidence:** Doc 1 (Title: Warner Records) Warner Bros. Records Parent company Warner Music Group Founded March 19, 1958...

**a2:** Warner Music Group

**Reviewer:** Accept

**Final Answer:** Warner Music Group

Figure 11: Although the initially decomposed sub-question still contains a compounded expression, CloRe is able to retrieve candidate documents covering the required evidence, showing that the retrieval process can still recover relevant support in this case. For the first sub-question, the initial attempt fails at all evidence stages; after sub-question rewriting and re-retrieval, CloRe finally recovers the correct staged answer and proceeds to solve the remaining step. Purple text indicates failed sub-questions or intermediate answers.

## Case study for CloRe of backtracking over explicit dependencies

**Question:** What date did the Governor of the city where the basilica named after the same saint as the one that Mantua Cathedral is dedicated to end?

**Ground Truth:** ["1952"]

### Problem Planning

q1: To which saint is Mantua Cathedral dedicated? [Retrieve]

q2: Which city is the basilica dedicated to #1 located in? [Retrieve]

q3: On what date did the Governor of #2 end? [Retrieve]

### Reasoning Steps

#### Step 1

**q1: To which saint is Mantua Cathedral dedicated?**

**Key Units:** ["Mantua Cathedral"]

**Stage:** Stage 0

**Selected Evidence:** Doc 1(Title: Mantua Cathedral) [Mantua Cathedral](#) (Italian: Cattedrale di San Pietro apostolo; Duomo di Mantova) in Mantua, Lombardy, northern Italy, is a Roman Catholic cathedral dedicated to Saint Peter...

**a1: Saint Peter**

**Reviewer:** Accept

#### Step 2

**Rewritten q2: Which city is the basilica dedicated to Saint Peter located in?**

**Key Units:** ["Saint Peter"]

**Stage:** Stage 0

**Selected Evidence:** Doc 1(Title: St. Peter's Basilica) [The Papal Basilica of St. Peter in the Vatican](#) (Italian: Basilica Papale di San Pietro in Vaticano), or simply St. Peter's Basilica (Latin: Basilica Sancti Petri), is an Italian Renaissance church in Vatican City, the papal enclave within the city of Rome.

**a1: Rome**

**Reviewer:** Accept

#### Step 3

**Rewritten q3: On what date did the Governor of Rome end?**

**Key Units:** ["Rome"]

Stage 0 → a3: Unknown (Reject); Stage 1 → a3: Unknown (Reject);

Stage 2 → a3: Unknown (Reject); Stage 3 → a3: Unknown (Reject)

**Rewritten q3 (again): On what date did the Governor of Rome end?**  
**(still fails!!! → Action: backtrack to q2)**

#### Step 4

**Rewritten q2: In which city is the basilica dedicated to Saint Peter located?**

Previous answer "Rome" marked as unreliable

**Key Units:** ["Saint Peter"]

**Stage:** Stage 0

**Selected Evidence:** Doc 1(Title: St. Peter's Basilica) [The Papal Basilica of St. Peter in the Vatican](#) (Italian: Basilica Papale di San Pietro in Vaticano), or simply St. Peter's Basilica (Latin: Basilica Sancti Petri), is an Italian Renaissance church in Vatican City, the papal enclave within the city of Rome.

**a2: Vatican City**

**Reviewer:** Accept

#### Step 5

**q3: On what date did the Governor of the Vatican City end ?**

**Key Units:** ["Vatican City"]

**Stage:** Stage 0

**Selected Evidence:** Doc 1(Title: Governor of Vatican City) [The post of Governor of Vatican City](#) (Governatore dello Stato della Città del Vaticano in Italian) was held by Marchese Camillo Serafini from the foundation of the state in 1929 until his death in 1952.

**a3: 1952**

**Reviewer:** Accept

**Final Answer:** 1952

Figure 12: CloRe's backtracking process over explicit dependencies in this case. The system first completes forward reasoning and obtains an intermediate answer chain. However, under this dependency, the downstream query about the Governor of Rome still fails to pass review after all evidence stages and an additional rewrite. CloRe therefore attributes the failure to its explicit predecessor and backtracks to re-solve Q2, revising the previous answer *Rome* to *Vatican City*. Under the corrected dependency, the system re-executes the remaining sub-question and produces the final answer *1952*. Purple text denotes failed sub-questions or intermediate answers.

---

**Prompt for dependency-aware question decomposition**

---

You are a planner for multi-hop question answering. Your task is to decompose the input question into the shortest executable sequence of sub-questions.

**Rules**

1. Output an ordered list of sub-questions using the format “### Q{i}:”.
2. When a later step depends on an earlier result, explicitly refer to it with “#i”. Do not introduce unused intermediate results.
3. Keep the decomposition concise and executable. Do not generate redundant verification, background explanation, paraphrases, or unnecessary extra steps.
4. For each sub-question, output “### Need Context: Yes” if it requires external knowledge retrieval; output “### Need Context: No” only if the sub-question can be answered solely from previously obtained results without any external retrieval, such as comparison, arithmetic, sorting, or simple logical composition.
5. The final sub-question must directly target the answer to the original question.

**Example 1**

## Question:

Tekken Comic has been published in the online version of the monthly manga magazine published by whom?

## Decomposed Question:

### Q1: Which monthly manga magazine published “Tekken Comic” in its online version?

### Need Context: Yes

### Q2: Who is the publisher of magazine #1?

### Need Context: Yes

**Example 2**

## Question:

A member of the FOLIO 40 and an early donor to the Donald Trump Presidential Campaign is the CEO of a media organization in what city in Florida?

## Decomposed Question:

### Q1: Who is a member of the FOLIO 40 and an early donor to the Donald Trump Presidential Campaign?

### Need Context: Yes

### Q2: Which media organization does #1 serve as CEO?

### Need Context: Yes

### Q3: In what city in Florida is the headquarters of #2 located?

### Need Context: Yes

**Now solve the following question.**

## Question:

{question}

## Decomposed Question:

---

Table 18: Prompt for dependency-aware question decomposition

---

**Prompt for conditional query rewriting**

---

You are a query rewriting assistant for a multi-hop RAG system. Your task is to rewrite the current sub-question into a self-contained and retrieval-ready query when dependency information from previous steps is needed.

**Rules**

1. Rewrite the current sub-question by replacing each dependency placeholder (e.g., #i) with the corresponding answer from the previous QA history.
2. Keep the interrogative intent consistent with the original sub-question.
3. Inject only the minimal dependency-resolving context needed to make the rewritten query self-contained.

**Example 1**

Key\_Question: When was the director of 39 East born?

Previous\_QA\_History: Q1: Who is the director of 39 East?, #1: John S. Robertson

Modifiable\_Question: When was #1 born?

Modified\_Question: When was the director John S. Robertson born?

**Example 2**

Key\_Question: What is the capital of the state where the Space Needle is located?

Previous\_QA\_History: Q1: In which U.S. state is the Space Needle located?, #1: Washington

Modifiable\_Question: What is the capital of #1?

Modified\_Question: What is the capital of the U.S. state Washington?

**Input Data**

Key\_Question: {key\_question}

Previous\_QA\_History: {history}

Modifiable\_Question: {sub\_question}

**Model output**

Modified\_Question:

---

Table 19: Prompt for conditional query rewriting.

---

**Prompt for failure-driven retry rewriting**

---

You are a query rewriting assistant for a multi-hop RAG system. The current sub-question has failed to retrieve sufficient evidence for reliable answering. Your task is to rewrite it into a query better suited for retrieval while preserving its original intent.

**Rules**

1. Keep the interrogative intent consistent with the original sub-question.
2. Use the original question and previous QA history to add only the minimal disambiguating context needed to make the query more retrieval-effective.
3. Rephrase the failed sub-question into a more search-friendly form without introducing unsupported assumptions or irrelevant details.

**Example**

Original\_Question: When was the director of 39 East born?

QA\_History: Q1: Who is the director of 39 East?, #1: John S. Robertson

Failed\_Subquestion: When was John S. Robertson born?

Rewritten\_Query: What is the birth date of the director John S. Robertson?

**Input Data**

Original\_Question: {original\_question}

QA\_History: {history}

Failed\_Subquestion: {subquestion}

**Model output**

Rewritten\_Query:

---

Table 20: Prompt for failure-driven retry rewriting.

---

**Prompt for step-level review**

---

You are a reasoning reviewer for multi-hop question answering. Your task is to evaluate the current candidate sub-answer using the current sub-question, the retrieved evidence, and the dependency-relevant reasoning history.

**Rules**

1. Determine whether the candidate sub-answer is supported by the retrieved evidence and remains consistent with the reasoning history. Pay particular attention to entity identity, dates, numbers, proper nouns, and relation consistency.

2. If the candidate answer is sufficiently supported and consistent, return accept.

If it is incorrect but can be reliably corrected from the provided evidence, return revise and provide the corrected answer.

If the evidence is insufficient, contradictory, or unable to support a reliable correction, return reject.

3. Do not introduce unsupported facts or change the target of the current sub-question. Return only a JSON object in the required format.

**Input Data**

Original\_Question: {original\_query}

Reasoning\_History: {reasoning\_history}

Current\_Subquery: {current\_sub\_query}

Retrieved\_Evidence: {retrieved\_context}

Candidate\_Subanswer: {current\_sub\_answer}

**Model output**

{"status": [accept | revise | reject], "revised\_answer": [corrected answer if status = revise; otherwise empty string]}

---

Table 21: Prompt for step-level review.

---

**Prompt for answering retrieval-required sub-questions**

---

You are a question answering assistant for multi-hop RAG. Given a sub-question and its retrieved context, produce a concise answer to the sub-question. If the answer cannot be determined reliably, output [UNKNOWN].

**Input Data**

Subquestion: {subquestion}

Context: {context}

**Model output**

[concise answer]

---

Table 22: Prompt for answering retrieval-required sub-questions.

---

**Prompt for answering non-retrieval sub-questions**

---

You are a reasoning assistant for multi-hop question answering. Answer the current sub-question using only the provided QA history. Do not use external knowledge or introduce unsupported facts. Return only the answer, with no explanation or extra text.

**Input Data**

Previous\_QA\_History: {history}

Current\_Subquestion: {subquestion}

**Model output**

[concise answer]

---

Table 23: Prompt for answering non-retrieval sub-questions.

---

**Prompt for final answer aggregation**

---

You are an answer aggregator for multi-hop question answering. Use only the provided context, which consists of the completed sub-questions and their corresponding sub-answers, to produce the final answer to the original question.

**Input Data**

Original\_Question: {original\_question}

Sub\_questions\_and\_answers: {subquestions\_and\_answers}

**Model output**

Answer: [final concise answer]

---

Table 24: Prompt for final answer aggregation.