CIRAG: Construction-Integration Retrieval and Adaptive Generation for Multi-hop Question Answering

Zili Wei, Yilin Wang, Xiaocui Yang*, Shi Feng*,
Weidong Bao, Daling Wang, Zihan Wang, Yifei Zhang
[1]School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China
weizl2@mails.neu.edu.cn {fengshi,yangxiaocui}@cse.neu.edu.cn

Abstract

Triple-based Iterative Retrieval-Augmented Generation (iRAG) mitigates document-level noise for multi-hop question answering. However, existing methods still face limitations: (i) **greedy single-path expansion**, which propagates early errors and fails to capture parallel evidence from different reasoning branches, and (ii) **granularity-demand mismatch**, where a single evidence representation struggles to balance noise control with contextual sufficiency. In this paper, we propose the Construction-Integration Retrieval and Adaptive Generation model, **CIRAG**[1]. It introduces an Iterative Construction-Integration module that constructs candidate triples and history-conditionally integrates them to distill core triples and generate the next-hop query. This module mitigates the greedy trap by preserving multiple plausible evidence chains. Besides, we propose an Adaptive Cascaded Multi-Granularity Generation module that progressively expands contextual evidence based on the problem requirements, from triples to supporting sentences and full passages. Moreover, we introduce Trajectory Distillation, which distills the teacher model's integration policy into a lightweight student, enabling efficient and reliable long-horizon reasoning. Extensive experiments demonstrate that CIRAG achieves superior performance compared to existing iRAG methods.

1 Introduction

Retrieval-Augmented Generation (RAG) excels in simple queries (Lewis et al., 2020; Lin et al., 2024; Ram et al., 2023) but struggles with multi-hop reasoning (Trivedi et al., 2023; Fan et al., 2024; Mallen et al., 2023), as single-step retrieval often fails to gather interconnected evidence (Shao et al., 2023). Iterative RAG (iRAG) (Trivedi et al., 2023; Asai et al., 2024; Yao et al., 2025) is introduced by retrieving information in multiple steps. However, existing iRAG methods, whether retrieving full documents (Zhao et al., 2021) or generating chain-of-thoughts (Trivedi et al., 2023), often accumulate irrelevant noise (Yoran et al., 2024) or factual hallucinations during iterations (Wang et al., 2023; Luo et al., 2024), which ultimately degrades reasoning reliability.

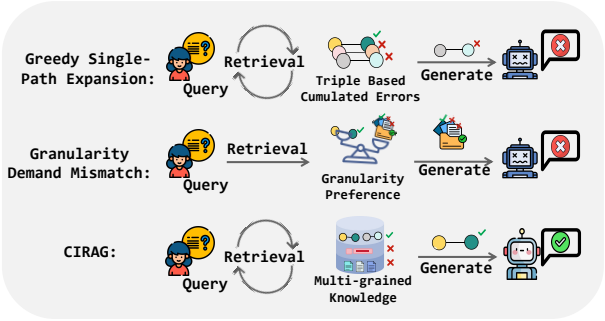

Figure 1: Challenges in Triple-based Retrieval.

To mitigate this issue, recent research has pivoted towards *triple-based retrieval* (Fang et al., 2025, 2024; Zhang et al., 2025). By using structured knowledge triples as retrieval units, these methods aim to achieve a more focused and reliable retrieval process. Despite these advances, current triple-based paradigms face two critical limitations, as illustrated in Figure 1. The first challenge is the **Greedy Single-Path Expansion** in retrieval. Existing methods predominantly select only the single best triple at each step (Fang et al., 2024, 2025). This linear strategy is inherently fragile: minor errors in early decisions can rapidly propagate and compound (Jiapeng et al., 2024; Shi et al., 2023; Lee et al., 2022). Moreover, by committing to a single path, it overlooks parallel evidence that is often essential for answering complex queries, resulting in incomplete or fragmented reasoning chains (Zhang et al., 2024). The second challenge is the **Granularity-Demand Mismatch**. Current

* Corresponding author..
[1]Our code can be found via https://github.com/52566rz/CIRAG.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 26181–26197

paradigms typically adopt a static evidence representation, failing to account for the heterogeneous information needs of different questions (Fang et al., 2024, 2025). For simple, relation-centric queries, retrieving full documents introduces substantial noise compared to concise triples. Conversely, for complex reasoning tasks that require rich contextual information, structured triples often discard crucial context (Wang and Han, 2025), omitting linguistic nuances that are naturally preserved in passages. As a result, a fixed retrieval granularity is insufficient to meet the diverse reasoning requirements posed by different queries.

To address these challenges, we draw inspiration from Construction-Integration (CI) model in cognitive psychology (Kintsch and Van Dijk, 1978; Wharton and Kintsch, 1991). CI characterizes human comprehension as a two-stage process. In Construction, semantic units are broadly activated, and in the Integration stage, contextual constraints suppress irrelevant activations to yield a coherent semantic network. Grounded in CI, we propose the Construction-Integration Retrieval and Adaptive Generation model, **CIRAG**, consisting of the Iterative Construction-Integration (ICI) Retrieval module and the Adaptive Cascaded Multi-Granularity Knowledge-Enhanced Generation (ACMG) module.

Specifically, ICI instantiates CI for iterative retrieval to mitigate greedy single-path expansion. At each iteration, the construction phase activates candidate triples extracted from retrieved documents, while the Integration phase enforces global constraints from the accumulated history to suppress noisy or off-path candidates, yielding a core triple set. Based on the uncovered knowledge gap, the integrator further synthesizes the next-hop query to continue the CI loop. By repeatedly alternating between construction and integration, ICI retains the coherent evidence network and reduces single-path bias. ACMG tackles the granularity-demand mismatch via dynamically expanding context, beginning with compact triples and escalating to supporting sentences or passages only when required, balancing noise control with contextual completeness. To ensure the core advantages of CIRAG are preserved even in smaller-scale models, we introduce Trajectory Distillation, which transfers integration trajectories from a strong teacher model to a lightweight student, enabling robust multi-step integration with reduced computational overhead. Our contributions can be summarized as follows:

- Inspired by the CI model, we propose CIRAG, which effectively addresses the challenges of greedy retrieval bias and mismatched granularity requirements in multi-hop reasoning by synergistically combining ICI and ACMG modules.
- We propose Trajectory Distillation to transfer integration trajectories from a teacher to an efficient student model, ensuring reasoning capabilities with reduced computational overhead.
- Extensive experiments on multiple multi-hop and single-hop QA benchmarks validate the effectiveness of CIRAG.

2 Related Works

2.1 Text-based iRAG

The current iRAG methods primarily retrieve relevant text passages from the corpus at each retrieval, providing context for LLM generation. IRCoT (Trivedi et al., 2023) and Iter-RetGen (Shao et al., 2023) dynamically generate sub-queries, retrieve relevant documents, and iteratively refine the reasoning trajectory throughout the generation process. FLARE (Jiang et al., 2023) focuses on adaptively retrieving documents when low-probability tokens are generated. MetaRAG (Zhou et al., 2024) first generates heuristic answers based on the question and the retrieved documents, and then refines them through retrieval. DualRAG (Cheng et al., 2025) guides retrieval through reason-driven query generation and integrates multi-round retrieval documents with an entity-centric approach. These models perform iterative retrieval by progressively augmenting the query with previously retrieved documents (Zhao et al., 2021; Trivedi et al., 2023). However, retrieved documents often include noise or irrelevant information (Yoran et al., 2024). The propagation of these distracting contexts can degrade retrieval quality and ultimately hinder overall RAG performance.

2.2 Triple-based iRAG

To reduce the impact of noise in documents, the triple-based iterative RAG method improves retrieval granularity by using knowledge triples during the retrieval process (Gutiérrez et al., 2024, 2025; Li et al., 2025). KiRAG (Fang et al., 2025) decomposes documents into structured triples and gradually expands the knowledge chain composed of triples during iterative retrieval, thereby accurately locating the key information missing in

multi-hop question answering. TeaRAG (Zhang et al., 2025) employs a triple-enhanced iterative retrieval strategy, simultaneously retrieving text blocks and pre-built triples in each iteration. However, existing methods face two challenges: the Greedy Single-Path Expansion and the Granularity-Demand Mismatch. Unlike existing methods, our approach addresses these challenges by obtaining a core set of triples through integration based on historical information in each iteration, and then cascadingly expanding the context to match the most appropriate granularity of information for each question.

3 CIRAG

3.1 Problem Formulation

We formally define the RAG task: given a user question x and a large-scale document corpus $D = \{d_i\}_{i=1}^{N}$, the objective of a RAG system is to generate an accurate answer $\hat{a}$ by retrieving and leveraging relevant documents from D.

3.2 Overview

As illustrated in Figure 2, we propose **CIRAG**, a two-module framework comprising Iterative Construction-Integration (ICI) and Adaptive Cascaded Multi-Granularity Knowledge-Enhanced Generation (ACMG). **ICI** retrieval module, retrieves the core triples set through two iterative phases: construction phase and integration phase. **ACMG** module generates the final answer by selecting the most appropriate information for the question through an adaptive cascading method.

3.3 Iterative Construction-Integration

Construction Phase. In the t iteration of ICI (see 1.1 in Figure 2), the retriever $\mathcal{R}$ retrieves the top-K documents most relevant to the current query a_t, forming a document set $\mathcal{D}_t = \{d_t^j\}_{j=1}^{K}$. These documents constitute the discourse context and are segmented into a unified sentence set $\mathcal{S}_t = \{s_t^k\}_{k=1}^{M}$. Leveraging a prompt-based approach, for each document in $\mathcal{D}_t$, we employ LLM to identify entities as intermediate anchors, directly guiding the extraction of relational triples(Edge et al., 2024; Fang et al., 2024). The prompt is provided in Appendix A.1. The retriever $\mathcal{R}$ as the reranker first ranks the triples by calculating their semantic similarity to the current query a_t. The top-N [2] ranked triples are selected to form the candidate triple set $\hat{\mathcal{T}}_t = \{\tau_t^i\}_{i=1}^{Q}$. To facilitate efficient context mapping, we explicitly record the provenance of each triple by constructing two mapping sets: $\mathcal{M}_t^{\mathcal{D}} : [Q] \to [K]$ and $\mathcal{M}_t^{\mathcal{S}} : [Q] \to [M]$, where $\mathcal{M}_t^{\mathcal{D}}(i) = j$ and $\mathcal{M}_t^{\mathcal{S}}(i) = k$ indicate that triple τ_t^i is extracted from document d_t^j and sentence s_t^k.

Integration Phase. In the integration phase of iteration t (see 1.2 in Figure 2), we employ a discriminative model $\mathcal{K}_D$ to filter candidate triples and generate the next query. In this phase, the model assesses the alignment of candidates with the current query a_t, while strictly anchoring to the original question x to maintain global consistency. Simultaneously, it synthesizes the historical context to identify information gaps to formulate a targeted query for the next iteration. To empower $\mathcal{K}_D$ with the capability to execute this complex dynamic reasoning, we optimize it via **Trajectory Distillation** (detailed in Sec. 4.3). Given the original question x, the initial candidate set $\hat{\mathcal{T}}_1$, an instruction prompt I provided in Appendix A.2, and the historical context $\mathcal{H}_{<t}$, $\mathcal{K}_D$ produces (i) a reasoning trace r_t, (ii) a filtered core triple set $\tilde{\mathcal{T}}_t$, and (iii) the next-round query a_{t+1}:

$$(r_t, \tilde{\mathcal{T}}_t, a_{t+1}) = \mathcal{K}_D(x, \hat{\mathcal{T}}_1, I, \mathcal{H}_{<t}). \quad (1)$$

The historical context $\mathcal{H}_{<t}$ encapsulates the reasoning trajectory of previous iterations, including the model output of each round and the corresponding candidate triples. Specifically, for $\mathcal{H}_{<1} = \varnothing$, for $t > 1$, it is defined as:

$$\mathcal{H}_{<t} = \left\{(r_i, \tilde{\mathcal{T}}_i, a_{i+1}, \hat{\mathcal{T}}_{i+1})\right\}_{i=1}^{t-1}. \quad (2)$$

The core triple set $\tilde{\mathcal{T}}_t$ retains only salient information pertinent to the query. To facilitate multi-granularity reasoning, we project these triples back to their source contexts via the provenance mappings, deriving the core sentence set $\tilde{\mathcal{S}}_t$ and document set $\tilde{\mathcal{D}}_t$:

$$\tilde{\mathcal{S}}_t = \{s_k \in \mathcal{S}_t \mid (i,k) \in \mathcal{M}_t^{\mathcal{S}}, \tau_i \in \tilde{\mathcal{T}}_t\}, \quad (3)$$

$$\tilde{\mathcal{D}}_t = \{d_j \in \mathcal{D}_t \mid (i,j) \in \mathcal{M}_t^{\mathcal{D}}, \tau_i \in \tilde{\mathcal{T}}_t\}. \quad (4)$$

Finally, we update the cumulative triple set $\mathbb{C}$, cumulative sentence set $\mathbb{S}$, and cumulative document set $\mathbb{D}$ via the incremental updates $\mathbb{C} \leftarrow \mathbb{C} \cup \tilde{\mathcal{T}}_t$, $\mathbb{S} \leftarrow \mathbb{S} \cup \tilde{\mathcal{S}}_t$, and $\mathbb{D} \leftarrow \mathbb{D} \cup \tilde{\mathcal{D}}_t$. Together, these accumulated sets $\{\mathbb{C}, \mathbb{S}, \mathbb{D}\}$ constitute the multi-granularity context for cascaded generation.

This iterative process terminates when $a_{t+1} = \varnothing$ or the maximum iteration step L is reached.

[2] We provide analysis of the effect of N in Appendix C.2.

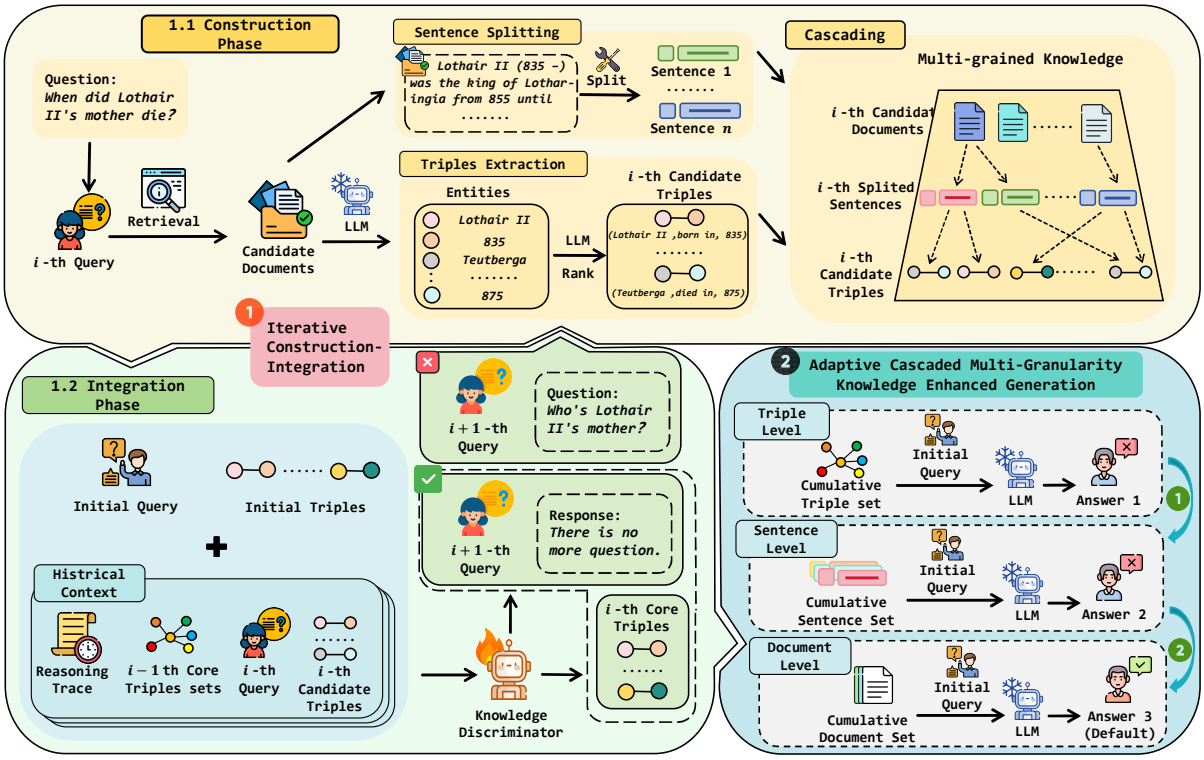

Figure 2: Overview of CIRAG. At each iteration, it employs an Iterative Construction-Integration (ICI) Retrieval module to retrieve a core triple set and record provenance links to their supporting sentences and documents, including two iterative phases: construction phase and integration phase. The core triple set is used to produce the final answer via the Adaptive Cascaded Multi-Granularity Knowledge-Enhanced Generation (ACMG) module.

Otherwise, a_{t+1} triggers the retriever to initiate the construction phase at round $t+1$. Through this cycle, the system progressively narrows the retrieval space while integrating verified knowledge for multi-hop reasoning.

3.4 Adaptive Cascaded Multi-Granularity Knowledge Augmented Generation

To fully leverage the complementary advantages of different granularities of knowledge in multi-hop question answering, we analyze their trade-offs in semantic expressiveness and noise control. Triples have clear structures and minimal noise, but often lack contextual information. Sentences enrich these triples through local context, thereby improving semantic integrity, but inevitably introducing some irrelevant information. Documents provide the most comprehensive global context, but have the most background noise. Existing approaches typically rely on a single granularity of evidence, making it difficult to balance semantic completeness and noise control.

To address this, we propose the Adaptive Cascaded Multi-Granularity Augmented Generation (ACMG) module (see 2 in Figure 2). We define the hierarchy of context granularities as an ordered sequence $\mathcal{G} = (g_{\mathbb{C}}, g_{\mathbb{S}}, g_{\mathbb{D}})$, where the precedence relation $\prec$, i.e., $g_{\mathbb{C}} \prec g_{\mathbb{S}} \prec g_{\mathbb{D}}$, denotes an increasing order of both semantic coverage and potential noise. This ordering allows the framework to prioritize high-precision, low-noise evidence and escalate to more exhaustive contexts only when the current level is insufficient.

Formally, for each granularity $g \in \mathcal{G}$, the model generates a response $a^{(g)}$ based on the question x, the corresponding context $C^{(g)}$, and a sufficiency instruction $I^{(g)}$ provided in Appendix A.3. This instruction directs the model $\mathrm{M_R}$ to evaluate the adequacy of $C^{(g)}$ relative to x. It produces a valid answer if the information is sufficient, or a predefined refusal response (e.g., Unanswerable) otherwise:

$$a^{(g)} = \mathrm{M_R}\left(x,\ C^{(g)},\ I^{(g)}\right) \tag{5}$$

where $C^{(g)}$ is the granularity-specific context selected from the accumulated pools $\{\mathbb{C}, \mathbb{S}, \mathbb{D}\}$ according to $g \in \mathcal{G}$.

To identify the minimal sufficient granularity, we define a sufficiency indicator $\mathrm{Suf}(a) \in \{0, 1\}$ that checks whether the model answer a is a refusal. Specifically, $\mathrm{Suf}(a) = 0$ if a matches a predefined refusal template, and $\mathrm{Suf}(a) = 1$ oth-

erwise. The system executes a cascaded search strictly following the precedence defined in $\mathcal{G}$, selecting the most concise yet sufficient granularity:

$$g^\star = \min_{\prec} \Big\{ g \in \mathcal{G} \mid \mathrm{Suf}\big(a^{(g)}\big) = 1 \Big\}. \quad (6)$$

The final output is the answer $a^{(g^\star)}$ from the first level that satisfies the sufficiency condition. If no level provides a sufficient answer, the system defaults to $a^{(g_{\mathrm{DOC}})}$ to maximize answerability.

3.5 Trajectory Distillation

CIRAG is compatible with LLMs of different parameter scales. However, the Integration Phase is non-trivial: at each iteration, the model must (i) filter noisy candidate triples ($\hat{\mathcal{T}}_t \rightarrow \tilde{\mathcal{T}}_t$) and (ii) synthesize a strategic next-hop query (a_{t+1}), both conditioned on the accumulated history from previous iterations. Large LLMs are generally more reliable at maintaining such long-horizon consistency, but invoking them in an interactive loop incurs substantial computational overhead. In contrast, lightweight models are more efficient but often fail to produce stable filtering and query-planning decisions. To achieve an efficient yet reliable integrator, we propose **Trajectory Distillation**. The core idea is to distill interactive trajectories produced by a strong teacher model into a lightweight student model $\mathcal{K}_D$(Eq. (1)), so that the student can reproduce the teacher's stepwise integration decisions at a lower cost.

Teacher trajectory generation. A trajectory ξ records a sequence of interaction steps. Given a question x, an initial candidate triple set $\hat{\mathcal{T}}_1$, and an instruction prompt I, the teacher generates:

$$\xi = \big\{(y_t, o_{t+1})\big\}_{t=1}^{L_\xi} \sim \pi_T(\cdot \mid x, \hat{\mathcal{T}}_1, I), \quad (7)$$

where at step t the teacher produces an integration decision:

$$y_t = (r_t, \tilde{\mathcal{T}}_t, a_{t+1}),$$

consisting of an integration rationale r_t, the filtered core triple set $\tilde{\mathcal{T}}_t$, and the next-hop query a_{t+1}. The retriever then returns the subsequent observation:

$$o_{t+1} = \hat{\mathcal{T}}_{t+1} = \mathcal{R}(a_{t+1}),$$

which serves as a candidate set for the next iteration.

Student supervision. Following prior works (Chen et al., 2023; Gou et al., 2023; Kang et al., 2025), we fine-tune the student to predict the teacher's integration decisions, while treating retrieval observations as context rather than supervision targets:

$$\min_\theta \; -\mathbb{E}_{\substack{x \sim \mathcal{D}_{\text{train}} \\ \tau \sim \pi_T(\cdot \mid x, I)}} \sum_{t=1}^{L} \log \mathcal{K}_D\big(y_t \mid x, I, \tau_{<t}; \theta\big) \quad (8)$$

where $\tau_{<t} = \{(y_i, o_i)\}_{i=1}^{t-1}$. $\mathcal{D}_{\text{train}}$ denotes the training set, $\pi_T(\cdot \mid x, I)$ is the teacher trajectory distribution conditioned on input x and prompt I, $\mathcal{K}_D(\cdot; \theta)$ is the student model parameterized by θ, and L_τ is the length of trajectory τ. At each step t, the student predicts the reasoning trace r_t, the filtered triples t_t, and the next query a_t, conditioned on x and the trajectory history $\tau_{<t}$.

After distillation, $\mathcal{K}_D$ can execute the same integration loop by consistently selecting salient triples ($\tilde{\mathcal{T}}_t$) and synthesizing next-hop queries (a_{t+1}) across iterations, while remaining substantially more efficient than the teacher.

4 Experimental Setup

4.1 Datasets and Metrics

We evaluate our method on three multi-hop QA benchmarks: **HotpotQA** (Yang et al., 2018), **2WikiMultiHopQA**(**2WikiMQA**) (Ho et al., 2020), and **MuSiQue** (Trivedi et al., 2022). For each dataset, we randomly selected 1000 multi-hop questions for evaluation in the validation set as done in previous work (Trivedi et al., 2023). To create a more rigorous and realistic retrieval setting, we followed the IRCoT (Trivedi et al., 2023) settings and merged all supported and unsupported paragraphs from the selected questions in each dataset to build the retrieval database.

We evaluate the retrieval performance using **Recall@{3, 5}** (**R@{3, 5}**) as the metrics, following prior works (Trivedi et al., 2023; Gutiérrez et al., 2024). For QA performance, We use **Exact Match** (**EM**) and **F1** as evaluation metrics. More details can be found in Appendix B.

4.2 Baselines

We propose a simple and efficient Iterative Retrieval-Augmented Generation framework, which we mainly compare with the iterative RAG method. Specifically, it mainly includes the following categories of RAG methods: (i)

Method	Qwen2.5-7B						Qwen2.5-max					
	2WikiMQA		HotpotQA		MuSiQue		2WikiMQA		HotpotQA		MuSiQue	
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
NativeRAG	31.5	28.2	50.2	34.5	16.8	9.9	47.1	39.2	66.9	52.4	27.6	18.2
IRCOT	45.7	36.4	56.8	42.4	23.5	13.4	65.7	55.3	72.8	58.1	34.2	22.1
FLARE	43.1	35.2	56.4	41.9	23.8	13.5	-	-	-	-	-	-
MetaRAG	50.4	44.7	63.3	49.6	31.9	21.2	58.7	52.4	74.6	60.8	43.8	32.4
KiRAG	52.7	36.9	62.1	49.0	31.7	20.2	59.4	52.1	73.2	59.0	45.0	32.7
DualRAG	62.3	51.7	58.7	44.8	33.7	22.1	75.6	65.8	73.3	57.8	50.2	36.6
DualRAG-FT	65.6	53.8	62.6	47.1	35.8	25.1	-	-	-	-	-	-
Ours	**69.5**	**59.0**	**67.1**	**52.5**	**40.9**	**29.3**	**76.4**	**67.1**	**74.9**	**60.9**	**56.0**	**44.9**

Table 1: Results on three MHQA benchmarks with Qwen2.5-7B-Instruct and Qwen2.5-max as base LLMs. **Bold** marks the best and underline the second-best.

Method	2WikiMQA		HotpotQA		MuSiQue	
	F1	EM	F1	EM	F1	EM
CIRAG	**68.1**	**58.9**	**67.1**	**52.5**	**40.9**	**29.3**
w/o TD	59.7	49.8	62.5	48.8	33.1	22.2
w/o reranker	66.3	57.3	64.7	50.6	38.3	27.2

Table 2: Ablation Study on trajectory distillation and reranker for CIRAG Using Qwen2.5-7B-Instruct.

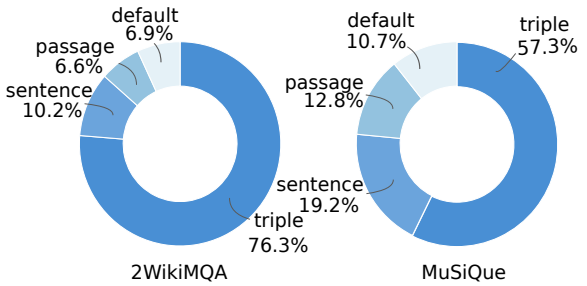

Figure 3: Context granularity distribution across datasets

NativeRAG (Lewis et al., 2020), which follows a retrieval-generation paradigm and generates answers based on documents retrieved once. (ii) Text-based iterative RAG methods, which iteratively retrieve relevant documents to gather the key information needed for multi-hop reasoning, such as IRCoT (Trivedi et al., 2023), FLARE (Jiang et al., 2023), MetaRAG (Zhou et al., 2024), DualRAG and its variant DualRAG-FT (Cheng et al., 2025). (iii) Triple-based iterative RAG method, which efficiently supplies the knowledge needed for multi-hop reasoning, using triples as retrieval units, such as KiRAG (Fang et al., 2025). More details can be found in Appendix B.3

4.3 Implementation and Training Details

Backbone. We use Qwen-2.5-7B-Instruct and Qwen-max-2025-01-25 (Yang et al., 2024) as the backbone of our framework and all baselines.

Retrieval Setup. We use two different retrieval models to validate the compatibility of our approach, including bge-Small-env1.5 (Xiao et al., 2024) and nvidia/NVEmbed-v2 (Lee et al., 2024). In the main experiment, all iterative RAG methods are set to a maximum of 4 iteration steps, and the retriever was used to retrieve the top 10 documents for each question for model inference.

Trajectory Distillation. Using Qwen-max-2025-01-25 as the teacher model, we apply CIRAG to generate 3,000 complete reasoning trajectories from the training sets of HotpotQA (Yang et al., 2018), 2WikiMQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022). We fine-tune Qwen-2.5-7B-Instruct as the student model utilizing Low-Rank Adaptation (Hu et al., 2022).

Further training configurations and implementation details are provided in Appendix B.4.

5 Results and Analysis

5.1 Main Results

Table 1 reports the main experimental results on three standard MHQA benchmarks. Overall, **CIRAG** consistently outperforms all baselines across varying model scales. We make the following key observations: (1) Compared with text-based iRAG methods, **CIRAG** improves performance by an average of 4.3% (F1) and 4.9% (EM) on Qwen2.5-7B-Instruct. Notably, KiRAG, despite optimizing only the triple-retrieval compo-

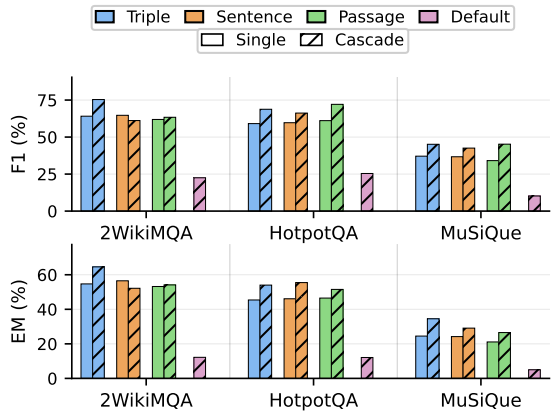

Figure 4: Comparison of single-granularity and cascaded performance

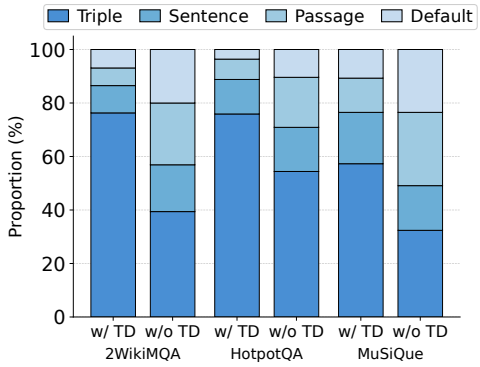

Figure 5: Effect of Trajectory Distillation on context granularity distribution

Method	2WikiMQA		HotpotQA		MuSiQue	
	F1	EM	F1	EM	F1	EM
CIRAG (Full)	**68.1**	**58.9**	**67.0**	**52.6**	**40.9**	**29.5**
w/o Passage	67.4	57.5	65.5	51.1	40.8	28.9
w/o Triple	67.5	58.8	64.9	51.0	38.7	26.9
w/o Sentence + Passage	64.1	54.7	59.1	45.4	37.1	24.5
w/o Triple + Passage	64.7	56.5	59.7	46.1	36.7	24.2
w/o Triple + Sentence	61.9	53.2	61.1	46.5	34.1	21.1

Table 3: Ablation Study on Cascaded Multi-Granularity Knowledge Augmented strategies for CIRAG Using Qwen2.5-7B-Instruct.

nent, remains competitive with stronger text-based approaches, e.g., MetaRAG and DualRAG. These results suggest that triples serve as finer-grained retrieval units than full passages, enabling more accurate, stable iterative retrieval, which ultimately benefits QA performance. (2) Compared with triple-based iRAG methods, **CIRAG** achieves an average gains of 10.3% (F1) and 11.6%(EM) with Qwen2.5-7B-Instruct. These improvements validate our design, yielding a better balance between structured precision and semantic completeness. The history-conditioned integration step mitigates the single-path bias of greedy linear expansion, while matching the context granularity to the question requirements. (3) Figure 3 reports the Granularity Distribution on 2WikiMQA and MuSiQue. We observe that triples dominate the cascade, accounting for 76.3% on 2WikiMQA and 57.3% on MuSiQue. This indicates that most multi-hop questions can be resolved with compact relational triples, allowing to avoid unnecessary noise. Meanwhile, the distribution shifts with dataset difficulty: in the more challenging MuSiQue dataset, the usage of coarser granularities is higher compared to 2WikiMQA. This contrast highlights the heterogeneous granularity demanded by different questions. Overall, these two observations confirm the rationale of our method, which can remain at low-noise triples when sufficient, yet reliably escalate to sentences or passages when additional context is required. (4) To verify the effectiveness of CIRAG, we report additional results under different retrievers and backbones in Appendix C.1.

5.2 Ablation Study

Effect of Evidence Granularity and Cascaded Context Expansion Table 3 compares different evidence granularities and cascade variants on Qwen2.5-7B-Instruct. We observe strong task dependency for single-granularity settings. On 2WikiMQA and MuSiQue, w/o Triple + Sentence underperform the others, suggesting that longer contexts introduce distracting noise; in contrast, passages are more effective on HotpotQA, indicating a higher need for paragraph-level context to recover bridging evidence. This contrast highlights heterogeneous granularity demands in multi-hop QA. Across all datasets, two-level cascades consistently outperform single-granularity baselines, showing that cascading can expand context when necessary while avoiding redundant context injection. CIRAG achieves the best overall F1/EM, confirming the benefit of full cascaded expansion. We further compare performance at each cascade stage with its single-granularity counterpart in Figure 4 . The cascade improves F1/EM consistently at every granularity, indicating that on-demand expansion selects more suitable evidence for different questions and yields reliable QA performance.

Effect of Trajectory Distillation To assess the impact of Trajectory Distillation (TD) on the *Integration Phase*, we implement a variant, w/o TD, where the distilled model $\mathcal{K}_D$ is replaced by a

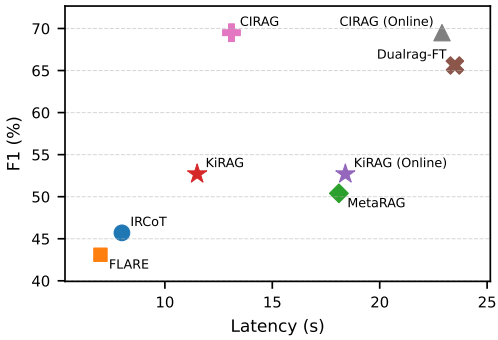

Figure 6: Latency vs. F1 on 2WikiMQA Using Qwen2.5-7B-Instruct.

Method	NQ		WebQ	
	F1	EM	F1	EM
NativeRAG	59.6	43.7	42.1	27.8
IRCOT	56.7	41.0	41.6	25.0
FLARE	56.3	43.0	42.3	26.0
MetaRAG	61.1	47.8	48.2	33.8
KiRAG	57.1	41.7	44.6	27.5
DualRAG	58.2	44.2	45.1	29.5
DualRAG-FT	60.2	45.7	47.4	31.2
CIRAG	**61.4**	**46.0**	**48.3**	**34.0**

Table 4: Additional results on single-hop QA datasets using Qwen2.5-7B-Instruct.

Method	Triple		Sentence		Passage	
	R@3	R@5	R@3	R@5	R@3	R@5
NVEmbed-v2	38.3	49.5	54.5	63.3	69.5	75.0
KiRAG	66.3	71.3	72.5	81.5	77.5	84.0
CIRAG	**71.5**	**82.3**	**78.3**	**89.0**	**84.5**	**91.5**

Table 5: Comparison of retrieval effectiveness across multiple granularities.

frozen Qwen-2.5-7B base model. As shown in Table 2, removing TD results in a significant degradation in QA performance. The decline is further elucidated by the granularity distribution in Figure 5. The utilization ratio of triples in the cascaded process markedly decreases, while the incidence of passages and default (where no matching granularity is found) surges. This implies a substantial deterioration in the quality of the core triple set retrieved by the ICI loop, which directly impairs downstream generation. These findings substantiate the critical role of TD in endowing the model with the capability to discriminate core triples and orchestrate multi-hop queries.

Effect of Reranker To assess the impact of the reranker, we introduce a variant, w/o reranker, which treats all triples extracted from retrieved documents as candidate triples. As shown in Table 2, this variant achieves comparable performance across all datasets, validating the effectiveness of the *Integration Phase* in identifying relevant triples.

5.3 Effect of the number of Iterations

Figure 7 shows the QA results of CIRAG and the iRAG baseline on the 2WikiMQA test set across varying maximum iteration steps L. CIRAG exhibits consistent gains from $L = 2$ to $L = 4$, demonstrating its ability to effectively accumulate multi-hop evidence. Notably, as L increases, baseline methods exhibit performance instability due to noise accumulation; in contrast, the performance of CIRAG smoothly plateaus beyond $L = 4$. This suggests that the ICI module effectively distills salient evidence without being affected by over-expansion. Furthermore, CIRAG achieves strong performance as early as $L = 2$, indicating that high-quality evidence is acquired in the initial retrieval stages, and it consistently outperforms the baseline across all values of L. Overall, these results demonstrate the effectiveness of the ICI module, highlighting both its stable convergence behavior and its efficiency.

5.4 Retrieval Performance of ICI module

To evaluate the retrieval effectiveness of the ICI module, we randomly sample 100 questions from the 2WikiMQA test set and manually annotate the essential knowledge triples required to answer each question, establishing them as the ground-truth triples. Utilizing these annotations alongside the gold passages and supporting sentences provided by the benchmark, we employ R@K to assess retrieval performance comprehensively across multiple granularities.

We compare the ICI module against NV-Embed-v2 and KiRAG. As shown in Table 5, ICI consistently outperforms KiRAG across all granularities. Notably, it yields substantial improvements at the triple level (+5.2% in R@3 and +11.0% in R@5), directly enhancing the model's capacity for structured reasoning. Coupled with the observed gains in downstream QA performance, these results suggest that the overall superiority of CIRAG is primarily driven by the ICI modules enhanced evidence acquisition and integration capabilities.

5.5 Efficiency Analysis

We evaluate the efficiency of CIRAG compared to baseline methods. To ensure a fair comparison, both CIRAG and the baselines utilize the identical retriever for document retrieval and employ the same underlying model as the reasoning component. Drawing inspiration from prior work (Fang et al., 2025), CIRAG extracts and caches knowledge triples offline to minimize online computational overhead. To quantify the impact of these pre-computed triples, we introduce a variant named CIRAG (Online), which performs dynamic triple extraction during the iterative retrieval process. Hardware configurations.

Figure 6 illustrates the average inference latency versus F1 score on the 2WikiMQA test set. The results indicate that: (1) Compared to CIRAG (Online), the offline pre-computation significantly reduces inference latency with negligible performance degradation, validating the efficiency benefits of our caching strategy. (2) CIRAG achieves higher F1 scores while maintaining relatively low latency, demonstrating that our approach enhances QA accuracy without introducing substantial computational costs. (3) Overall, CIRAG strikes a superior balance between effectiveness and efficiency, characterized by lower latency and higher F1 scores compared to baselines.

5.6 Other QA Tasks

To evaluate generalization, we conducted additional experiments on a multi-hop QA dataset, WebQA (Berant et al., 2013), and a single-hop QA dataset, NQ (Kwiatkowski et al., 2019). As shown in Table 4, CIRAG outperforms all baselines on WebQA and NQ, demonstrating strong generalization across QA settings.

5.7 Case Study

We conduct a case study in Appendix D to verify the effectiveness of our method.

6 Conclusion

We propose CIRAG, a two-module iRAG framework for multi-hop QA that integrates triple-based retrieval with adaptive multi-granularity generation. An Iterative Construction-Integration module distills core triples and plans next-hop queries to mitigate greedy reasoning and retrieval noise, while an Adaptive Cascaded Generation module dynamically expands context from triples to sentences and passages as needed. We further introduce Trajectory Distillation to enhance the integration capability of lightweight models. Experiments on MHQA benchmarks demonstrate consistent improvements over iRAG baselines. In the future, we will explore lightweight routing for efficient granularity selection.

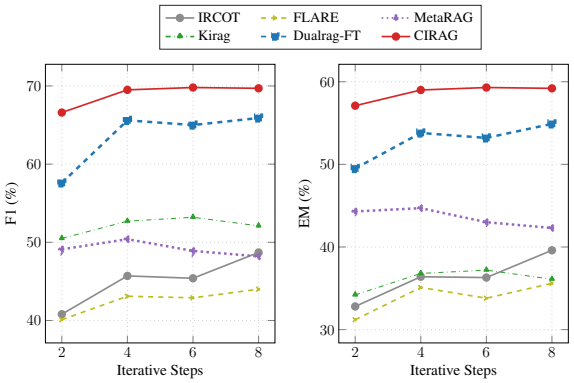

Figure 7: The effect of the number of iterative steps L for different models on the 2WikiMQA test set.

Limitations

Our framework has two main limitations. First, the Construction Phase depends on prompt-based open IE; in specialized domains or when relations are highly implicit, the extractor may miss key triples, limiting downstream integration. Future work could improve extraction via domain-adaptive training or prompt optimization. Second, the cascaded generation incurs extra latency due to sequential granularity checks. While Trajectory Distillation reduces integration cost, generation overhead remains. A potential improvement is to develop a lightweight granularity router that directly selects the appropriate context level, which could further accelerate inference.

Acknowledgments

The work is supported by the National Natural Science Foundation of China (Nos. 62272092, 62172086), and the Fundamental Research Funds for the Central Universities under Grants (N25XQD004).

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544.

Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. 2023. Fireact: Toward language agent fine-tuning. *arXiv preprint arXiv:2310.05915*.

Rong Cheng, Jinyi Liu, Yan Zheng, Fei Ni, Jiazhen Du, Hangyu Mao, Fuzheng Zhang, Bo Wang, and Jianye Hao. 2025. Dualrag: A dual-process approach to integrate reasoning and retrieval for multi-hop question answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31877–31899.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 6491–6501.

Jinyuan Fang, Zaiqiao Meng, and Craig MacDonald. 2024. TRACE the evidence: Constructing knowledge-grounded reasoning chains for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 8472–8494.

Jinyuan Fang, Zaiqiao Meng, and Craig Macdonald. 2025. Kirag: Knowledge-driven iterative retriever for enhancing retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18969–18985.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Tora: A tool-integrated reasoning agent for mathematical problem solving. *arXiv preprint arXiv:2309.17452*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Bernal J Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*.

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.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

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

Li Jiapeng, Liu Runze, Li Yabo, Zhou Tong, Li Mingling, and Chen Xiang. 2024. Tree of reviews: A tree-based dynamic iterative retrieval framework for multi-hop question answering. *arXiv preprint arXiv:2404.14464*.

Minki Kang, Jongwon Jeong, Seanie Lee, Jaewoong Cho, and Sung Ju Hwang. 2025. Distilling llm agent into small models with retrieval and code tools. *arXiv preprint arXiv:2505.17612*.

Walter Kintsch and Teun A Van Dijk. 1978. Toward a model of text comprehension and production. *Psychological review*, 85(5):363.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

Hyunji Lee, Sohee Yang, Hanseok Oh, and Minjoon Seo. 2022. Generative multi-hop retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1417–1436.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Rui Li, Quanyu Dai, Zeyu Zhang, Xu Chen, Zhenhua Dong, and Ji-Rong Wen. 2025. Knowtrace: Bootstrapping iterative retrieval-augmented generation with structured knowledge tracing. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 1470–1480.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, and 1 others. 2024. RA-DIT: Retrieval-augmented dual instruction tuning. In *International Conference on Learning Representations*.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*.

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

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

Matthew Renze. 2024. The effect of sampling temperature on problem solving in large language models. In *Findings of the association for computational linguistics: EMNLP 2024*, pages 7346–7356.

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

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop questions via single-hop question composition. *Trans. Assoc. Comput. 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.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, and 1 others. 2023. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*.

Jingjin Wang and Jiawei Han. 2025. Proprag: Guiding retrieval with beam search over proposition paths. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6223–6238.

Cathleen Wharton and Walter Kintsch. 1991. An overview of construction-integration model: a theory of comprehension as a foundation for a new cognitive architecture. *ACM Sigart Bulletin*, 2(4):169–173.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. 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.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Liu Weichuan, Lei Hou, and Juanzi Li. 2025. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27022–27043.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *International Conference on Learning Representations*.

Chao Zhang, Yuhao Wang, Derong Xu, Haoxin Zhang, Yuanjie Lyu, Yuhao Chen, Shuochen Liu, Tong Xu, Xiangyu Zhao, Yan Gao, and 1 others. 2025. Tearag: A token-efficient agentic retrieval-augmented generation framework. *arXiv preprint arXiv:2511.05385*.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Liu Yong, and Shen Huang. 2024. End-to-end beam retrieval for multi-hop question answering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1718–1731.

Chen Zhao, Chenyan Xiong, Jordan L. Boyd-Graber, and Hal Daumé III. 2021. Multi-step reasoning over unstructured text with beam dense retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4635–4641.

Yujia Zhou, Zheng Liu, Jiajie Jin, Jian-Yun Nie, and Zhicheng Dou. 2024. Metacognitive retrieval-augmented large language models. In *Proceedings of the ACM Web Conference 2024*, pages 1453–1463.

A Prompts

A.1 Prompt for Knowledge Triple Extraction

The prompt used for NER from a document is illustrated in Figure 9. The prompt used for extracting knowledge triples from a document is illustrated in Figure 10.

A.2 Prompt for Integration Phase

The prompt used for distilling a core triple set and synthesizing the next-hop query in the integration phase is illustrated in Figure 11.

A.3 Prompt for ACMG

Figure 12 illustrates the prompting method at the ACMG triplet level. The sentence and article levels differ from the triplet level only in the examples used.

B Experimental Details

B.1 Datasets

In our experiments, we employ four multi-hop QA datasets: HotpotQA, 2WikiMultiHopQA, MuSiQue, WebQuestions (WebQA), and one single-hop QA dataset: Natural Questions (NQ). For HotpotQA, 2WikiMultiHopQA, and MuSiQue, we construct the retrieval corpus by following exactly the same procedure as (Trivedi et al., 2023). For WebQA and NQ, we use the corpus version released with DPR. To control evaluation cost, for each question in WebQA and NQ, we include all annotated supporting documents and additionally sample up to 10 non-supporting documents.

For datasets with public test sets (WebQA and NQ), we randomly sample 500 test questions for evaluation. For datasets without public test sets (HotpotQA, 2WikiMultiHopQA, and MuSiQue), we randomly sample 1,000 questions from the development set as our test split and report performance on this subset. Since these three datasets are also used for training, we further randomly sample 1,000 questions from each original training set to form the training split used in our experiments.

B.2 Metrics Details

Exact Match (EM) provides the strictest criterion, assigning a score of 1 only when the predicted answer exactly matches the ground truth and 0 otherwise.

The F1 score measures token-level similarity by computing the harmonic mean of Precision and Recall, where Precision reflects the proportion of predicted tokens that are correct, and Recall denotes the proportion of reference tokens successfully retrieved. We follow evaluation metrics from MuSiQue (Trivedi et al., 2022) to calculate F1 scores for the final answer.

B.3 Baselines

For IRCoT and FLARE, we use the implementations provided by DualRAG (Cheng et al., 2025). For other models, we adapt the official implementations released by the authors to match our experimental setting. For a fair comparison, CIRAG and all baselines use the same retriever to access the same corpus and share the same backbone LLM for reasoning and answer generation. For methods that require training but do not provide public checkpoints, we reproduce training by following the authors' released data-generation pipelines to construct the training set, and we use the same initial training split as CIRAG. When applicable, we align the training recipe (e.g., optimization settings and hyperparameters) with the original papers as closely as possible under our hardware constraints.

B.4 Training and Hyperparameter Details

Training Details. We fine-tune student models using parameter-efficient tuning with LoRA (rank 64) (Hu et al., 2022). models are fine-tuned for 2 epochs using a batch size of 2 and a learning rate of $2 * 10^{-4}$. Experiments are conducted using four NVIDIA A6000 48GB GPUs,with a total training time of approximately 3 hours.

Implementation and Hyperparameter Details. Throughout the experiments, we set the maximum number of iterative steps L to 4. The details of each component in our CIRAG are outlined as follows: For the *Retriever* model, we

use either bge-Small-env1.5 (Xiao et al., 2024) or nvidia/NVEmbed-v2 (Lee et al., 2024) to retrieve documents and rerank triples. The number of retrieved documents per iteration (i.e., K) is 10. The number of candidate triples per iteration (i.e., N) is 30. For the *Backbone* model, we try different LLMs, including Llama3 (Grattafiori et al., 2024), Qwen-2.5-7B-Instruct, and Qwen-max-2025-01-25 (Yang et al., 2024) as the backbone of our framework and all baselines. We set the temperature to 0 when calling the API of Qwen-max and use greedy decoding for other models to avoid random sampling (Renze, 2024). We mainly report the performance of using Qwen-2.5-7B-Instruct and Qwen-max-2025-01-25 (Yang et al., 2024) as the Backbone. Hardware configurations: All experiments are conducted on the same hardware environment equipped with an Intel Xeon Gold 6326 CPU (2.90GHz, 32 cores) and an NVIDIA RTX A6000 GPU.

C Additional Experimental Results and Analysis

C.1 Using Different Retrievers and Readers

To validate the effectiveness of CIRAG, we provide additional results using different retrievers and a backbone model. Specifically, we replace the nvidia/NVEmbed-v2 Retriever with bge-Small-env1.5 Retriever for retrieving documents from the corpus and the other components remain unchanged. The corresponding QA performance is presented in Table 7, respectively. The results are consistent with those obtained using thenvidia/NVEmbed-v2 Retriever, demonstrating the adaptability and effectiveness of our CIRAG across different retriever models.

To verify the applicability of our approach across different LLM architectures, we also conducted experiments on other open-source models (Llama-3-8B-Instruct). The results are shown in Table 6. These experimental results indicate that our method is also applicable to other LLM architectures, demonstrating robust performance across different models.

C.2 Effect of the Number of Candidate Triples

During the construction phase of CIRAG, the reranker retrieves the top-N triples most relevant to the current query, which are treated as the candidate triple set. To study the sensitivity to N, we vary N from 10 to 100. Figure 8 reports the QA performance under different N values, over 3,000 questions sampled from three multi-hop QA datasets (the same as in the main experiment). Overall, CIRAG exhibits low sensitivity to N: both F1 and EM remain largely stable, with fluctuations below 2% across the tested range. This robustness is consistent with our integration module, which refines the candidate set into a compact core triple set by leveraging the accumulated history, thereby mitigating the effect of noisy candidates. Considering cost and efficiency, we chose N = 30.

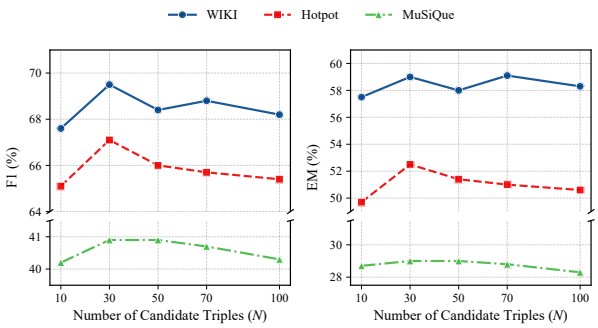

Figure 8: QA performance (%) of CIRAG under different values of N on three multi-hop QA datasets.

Method	2WikiMQA		HotpotQA		MuSiQue	
	F1	EM	F1	EM	F1	EM
IRCOT	41.1	23.1	56.7	41.3	23.1	13.6
FLARE	40.5	24.8	56.8	41.2	24.5	14.0
MetaRAG	44.9	30.6	<u>62.5</u>	<u>48.5</u>	31.7	20.8
Kirag	47.2	20.2	60.5	46.2	30.9	19.6
Dualrag	56.8	37.6	57.5	43.6	32.4	21.1
Dualrag-FT	<u>60.1</u>	<u>39.7</u>	61.9	46.0	<u>34.6</u>	<u>24.7</u>
Ours	**64.0**	**44.9**	**66.7**	**51.4**	**40.7**	**28.2**

Table 6: QA performance (%) using nvidia/NVEmbed-v2 as the Retriever and Llama-3-8B-Instruct as Backbone. The best and second-best performances are highlighted in bold and underlined, respectively.

C.3 Sensitivity to Trajectory Quantity in KD

To assess the sensitivity of KD to the number of training trajectories, we train KD models with 1k, 2k, 3k, and 4k trajectories under identical settings, following the trajectory construction procedure described in Sec. 4.3, and evaluate them on 100 randomly sampled instances from the test splits of HotpotQA, 2WikiMultiHopQA (2WikiMQA), and MuSiQue. We use Qwen-2.5-7B-Instruct as the backbone and nvidia/NVEmbed-v2 as the re-

Method	2WikiMQA		HotpotQA		MuSiQue	
	F1	EM	F1	EM	F1	EM
IRCOT	44.8	35.9	55.1	41.2	21.8	12.2
FLARE	41.2	34.4	54.1	39.6	21.3	11.8
MetaRAG	48.6	42.8	60.3	47.6	29.2	18.8
Kirag	51.9	36.9	61.5	<u>48.1</u>	30.7	19.5
Dualrag	61.2	51.1	57.6	44.5	32.9	21.0
Dualrag-FT	<u>64.5</u>	<u>52.3</u>	<u>61.7</u>	46.7	<u>35.2</u>	<u>24.8</u>
Ours	**66.8**	**57.3**	**65.6**	**51.8**	**39.9**	**27.8**

Table 7: QA performance (%) using bge-Small-env1.5 as the Retriever and Qwen2.5-7B-Instruct as Backbone. The best and second-best performances are highlighted in bold and underlined, respectively.

Quantity	2WikiMQA		HotpotQA		MuSiQue	
	F1	EM	F1	EM	F1	EM
1000	63.2	53.8	64.2	49.6	40.5	28.8
2000	65.9	55.8	65.5	50.4	41.7	30.4
3000	67.6	57.4	66.9	51.1	42.3	31.2
4000	68.4	57.5	67.5	51.6	42.8	31.0

Table 8: Effect of Training Trajectory Quantity on KD Performance.

trieval model. The QA results are shown in Table 8. The results indicate that even with only 1k trajectories, the model achieves strong performance, reaching F1 = 63.2 on 2WikiMultiHopQA, thereby demonstrating high sample efficiency. Performance improves consistently as the number of trajectories increases but exhibits diminishing returns beyond 3k. The gain from 3k to 4k trajectories amounts to only 0.8 F1 points on 2WikiMultiHopQA. Similar trends are observed across datasets.

These findings indicate that KD is not overly sensitive to trajectory quantity and that a moderate number of trajectories is sufficient to learn an effective integration policy.

C.4 Latency Analysis of the ACMG Module

Statistical results indicate that the ACMG module requires an average of only 1.3, 1.2, and 1.6 LLM calls per question on 2WikiMultiHopQA, HotpotQA, and MuSiQue, respectively. This demonstrates that ACMG entails few generation steps and infrequently resorts to document-level backtracking for inference. Owing to its efficient triplet-level reasoning mechanism, this cascaded architecture incurs negligible latency overhead.

D Case Study

We conducted several case studies to analyze the effectiveness of our method. The results show that DualRAG can dynamically retrieve information when additional knowledge is required, enabling efficient multi-hop reasoning with minimal iterations. Detailed case study examples are illustrated in Table 9.

Case Study

Question: Which film has the director who is older, God'S Gift To Women or Aldri Annet Enn Bråk?
Gold Answers: [God'S Gift To Women]

Iterative Construction-Integration (ICI) Retrieval module

Step 1
Construction Phase
Retrieved Documents
(1) God's Gift to Women: El extraño viaje is a 1964 Spanish black drama film directed by Fernando Fernán Gómez...
(2) Aldri annet enn bråk: Aldri annet enn bråk is a 1954 Norwegian comedy- drama film...
(3) Dan Milne: Dan Milne is a British actor/ director who is possibly best known for his role...
(4) ...

Candidate Triples
("god s gift to women", "directed by", "michael curtiz"), ("aldri annet enn br k", "directed by", "edith carlmar"), ("god s gift to women", "is a", "romantic musical comedy film")...

Integration Phase
Thought
The query asks: Which film has the director who is older, God's Gift to Women or Aldri Annét Enn Bråk? From the facts provided, we know that God's Gift to Women was directed by Michael Curtiz and Aldri Annét Enn Bråk was directed by Edith Carlmar. To determine which director is older, we need their birth years. The next step is to find out the birth years of Michael Curtiz and Edith Carlmar.

Core Triples
("god s gift to women", "directed by", "michael curtiz"), ("aldri annet enn br k", "directed by", "edith carlmar")

Next-hop query
What are the birth years of Michael Curtiz and Edith Carlmar?

Step 2
Construction Phase
Retrieved Documents
(1) Edith Carlmar: Edith Carlmar (Edith Mary Johanne Mathiesen) (15 November 1911 - 17 May 2003)...
(2) Michael Curtiz: Michael Curtiz (born Manó Kaminer(1886-1905) Mihály Kertész ...
(3) ...

Candidate Triples
("edith carlmar", "started", "carlmar film a s"), ("edith carlmar", "is also known as", "edith mary johanne mathiesen"), ("edith carlmar", "born on", "15 november 1911"), ("michael curtiz", "born on", "december 24 1886"),...

Integration Phase
Thought
The question is: Which film has the director who is older, God's Gift to Women or Aldri Annét Enn Bråk? From the previous step, we identified that Michael Curtiz directed God's Gift to Women and Edith Carlmar directed Aldri Annét Enn Bråk. The current facts provide their birth dates: Michael Curtiz was born on December 24, 1886 and Edith Carlmar was born on November 15, 1911. Since Michael Curtiz is older than Edith Carlmar. Therefore, the film God's Gift to Women has the older director.

Core Triples
("edith carlmar", "born on", "15 november 1911"), ("michael curtiz", "born on", "december 24 1886")

Next-hop query
<no question>

Adaptive Cascaded Multi-Granularity Knowledge-Enhanced Generation (ACMG) module

Cumulative Triple Set
("god s gift to women", "directed by", "michael curtiz"), ("aldri annet enn br k", "directed by", "edith carlmar"), ("edith carlmar", "born on", "15 november 1911"), ("michael curtiz", "born on", "december 24 1886")

Thought
Michael Curtiz (born Dec 24, 1886) is older than Edith Carlmar (born Nov 15, 1911). Therefore, the film with the older director is God'S Gift To Women.
Answer: God'S Gift To Women

Table 9: Case Study

Prompt Used for NER

Instruction: Your task is to extract named entities from the given paragraph. Respond with a JSON list of entities.

Examples:
Paragraph: *Radio City Radio City is India's first private FM radio station and was started on 3 July 2001. It plays Hindi, English and regional songs. Radio City recently forayed into New Media in May 2008 with the launch of a music portal - PlanetRadiocity.com that offers music related news, videos, songs, and other music-related features.*
Entity lists: *{"named entities": ["Radio City", "India", "3 July 2001", "Hindi", "English", "May 2008", "PlanetRadiocity.com"] }*

Inputs:
Passage: *{document text}*

Outputs:
Entity lists:

Figure 9: Prompt used for NER.

Prompt Used for Knowledge Triple Extraction

Instruction: Your task is to construct an RDF (Resource Description Framework) graph from the given passages and named entity lists. Respond with a JSON list of triples, with each triple representing a relationship in the RDF graph. Pay attention to the following requirements: - Each triple should contain at least one, but preferably two, of the named entities in the list for each passage. - Clearly resolve pronouns to their specific names to maintain clarity.

Examples: Paragraph: *Radio City Radio City is India's first private FM radio station and was started on 3 July 2001. It plays Hindi, English and regional songs. Radio City recently forayed into New Media in May 2008 with the launch of a music portal - PlanetRadiocity.com that offers music related news, videos, songs, and other music-related features.*
Entity lists: *{"named entities": ["Radio City", "India", "3 July 2001", "Hindi", "English", "May 2008", "PlanetRadiocity.com"] }*
Knowledge Triples: *{"triples": [["Radio City", "located in", "India"], ["Radio City", "is", "private FM radio station"], ["Radio City", "started on", "3 July 2001"], ["Radio City", "plays songs in", "Hindi"], ["Radio City", "plays songs in", "English"], ["Radio City", "forayed into", "New Media"], ["Radio City", "launched", "PlanetRadiocity.com"], ["PlanetRadiocity.com", "launched in", "May 2008"], ["PlanetRadiocity.com", "is", "music portal"], ["PlanetRadiocity.com", "offers", "news"], ["PlanetRadiocity.com", "offers", "videos"], ["PlanetRadiocity.com", "offers", "songs"]] }*

Inputs:
Title: *{document title}*
Text: *{document text}*

Outputs:
Knowledge Triples:

Figure 10: Prompt used for extracting knowledge triples.

Prompt Used for Integration Phase

Instruction: Your task is to loop through collecting facts to solve the query until enough facts are collected. You must plan to continue in a series of steps, looping as follows:[[## fact_before_filter ##]]{fact_before_filter}[[## thought ##]]{thought}[[## fact_after_filter ##]]{fact_after_filter} [[## question ##]] {question} sequence. In each step, in the [[## fact_before_filter ##]] section, this is the collection of facts that need to be filtered to solve the previous problem. In the [[## thought ##]] section, only use the information from the facts before filtering, briefly analyze the contribution of the facts to the problem, and list useful facts. Based on the analysis, determine whether the original problem can be solved. If it cannot be solved, analyze what additional information is needed and raise new questions. In the [[## fact_after_filter ##]] section, if there are no relevant facts, return an empty list:{ "fact": [] }.In the [[## question ##]] section, you need to use the filtered facts to propose the next problem that still needs to be solved. If the facts collected in the previous rounds are sufficient to answer the original query, output `<no question>` as the end to terminate the loop.

i-th Inputs:
[[## Original Query ##]]: *{query}*
[[## fact_before_filter ##]]: *{fact_before_filter}*
[[## historical_context ##]]: *{historical context}*

i-th Outputs:
[[## thought ##]]:
[[## fact_after_filter ##]]:
[[## question ##]]:

Figure 11: Prompt used for Integration Phase.

Prompt Used for triple level in ACMG.

Instruction: As an advanced reading comprehension assistant, your task is to carefully analyze triples and extract useful information from them to answer corresponding questions. Your response start after "Thought: ", where you will methodically break down the reasoning process, illustrating how you arrive at conclusions. If the provided information cannot answer the question, output Unanswerable. Conclude with "Answer: " to present a concise, definitive response, devoid of additional elaborations.

Examples: Triples: *triples: ('erik hort', 'born in', 'montebello'), ('erik hort', 'is', 'american'), ('montebello', 'is located in', 'rockland county')*
Query: *What county is Erik Hort's birthplace a part of?*
Thought: *The triplet ('erik hort', 'born in', 'montebello') tells us that Erik Hort was born in Montebello. The triplet ('montebello', 'is located in', 'rockland county') tells us that Montebello is located in Rockland County. Therefore, since Erik Hort was born in Montebello and Montebello is in Rockland County, Erik Hort's birthplace is part of Rockland County.*
Answer: *Rockland County.*

Inputs:
Triples: *{triples}*
Query: *{query}*

Outputs:
Thought:

Figure 12: Prompt used for triple level in ACMG.