

# To Retrieve or To Think? Cross-Boundary Context Evolution for Multi-hop Complex Reasoning

Rubing Chen<sup>1,3</sup>, Jian Wang<sup>2</sup>, Wenjie Li<sup>1</sup>, Xiao-Yong Wei<sup>2,1,3\*</sup>, Qing Li<sup>1,3</sup>

<sup>1</sup> Department of Computing, The Hong Kong Polytechnic University

<sup>2</sup> College of Computer Science, Sichuan University

<sup>3</sup> The Hong Kong Polytechnic University Shenzhen Research Institute

rubing.chen@connect.polyu.hk wangjian51@scu.edu.cn

cswjli@comp.polyu.edu.hk {cs007.wei, qing-prof.li}@polyu.edu.hk

## Abstract

Current context augmentation methods, such as retrieval-augmented generation, play a crucial role in bridging a model’s internal knowledge boundary and external evidence for multi-hop reasoning. However, they often follow a rigid policy and treat external retrieval as the default action at each step. Such brute-force context expansion incurs unnecessary computational cost and may degrade reasoning performance by saturating the context with redundant or weakly relevant evidence. In this paper, we propose cross-boundary Context Evolution (EvoCtx), a framework that models complex reasoning as an adaptive process of boundary-aware context evolution. EvoCtx dynamically decides whether the next reasoning transition should cross the current evidence boundary through retrieval or refine the reasoning state within the existing context. It estimates the semantic gap between the reasoning state and the accumulated evidence, and strategically alternates between boundary expansion and intra-boundary trajectory refinement. This eliminates redundant retrieval steps and preserves a compact, evidence-supported reasoning trajectory. Extensive experiments on challenging open-domain and multi-hop QA benchmarks demonstrate that EvoCtx significantly outperforms previous methods, offering an effective approach to complex reasoning tasks. The source code can be accessed at <https://github.com/AnyarB-Chen/EvoCtx>.

## 1 Introduction

Large language models (LLMs) solve complex tasks by conditioning their reasoning and generation on various contexts, which may contain instructions, demonstrations, or retrieved evidence (Brown et al., 2020; Wei et al., 2022). For open-domain complex tasks, retrieval-augmented generation (RAG) (Lewis et al., 2020) is particu-

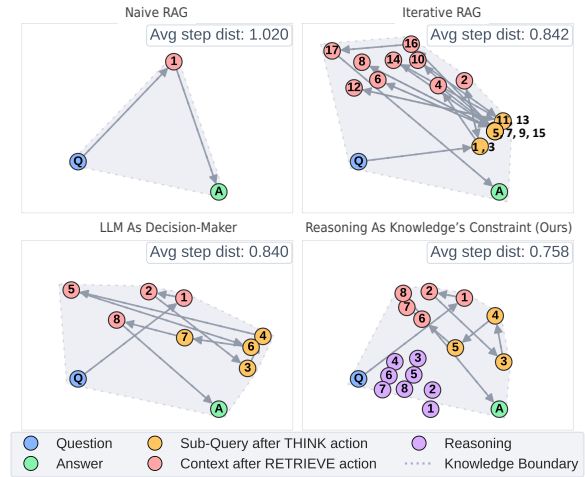


Figure 1: Visualized semantic trajectories of different retrieval-reasoning policies. Examples are selected from the MultiHop-RAG (Tang and Yang, 2024) dataset. Numbered nodes represent intermediate states. Unlike baselines that exhibit coarse jumps (Naive RAG), dispersed states (Iterative RAG), or unsupported steps (LLM as a decision-maker), our EvoCtx uses a boundary-aware retrieve-vs.-think controller to decide when to acquire new evidence and when to refine the search trajectory within the current evidence boundary, yielding a compact and evidence-supported trajectory.

larly important, as it bridges the gap between internal parametric knowledge and external evidence. However, many tasks require multiple hops and evidence is rarely identifiable in a single step. Later information needs often become visible only after earlier hops have been resolved (Press et al., 2023; Trivedi et al., 2023). Therefore, the key challenge is not merely how much context should be provided, but *how context should be acquired, updated, and controlled* as reasoning unfolds (Zhao et al., 2026; Wei and Yang, 2012b; Zhang et al., 2026).

Conventional context augmentation typically follows a “retrieve-then-generate” pipeline, implicitly assuming that the initial query can expose all evidence needed for the final answer (Press et al., 2023; Trivedi et al., 2023). This assumption is frag-

\*Corresponding author.

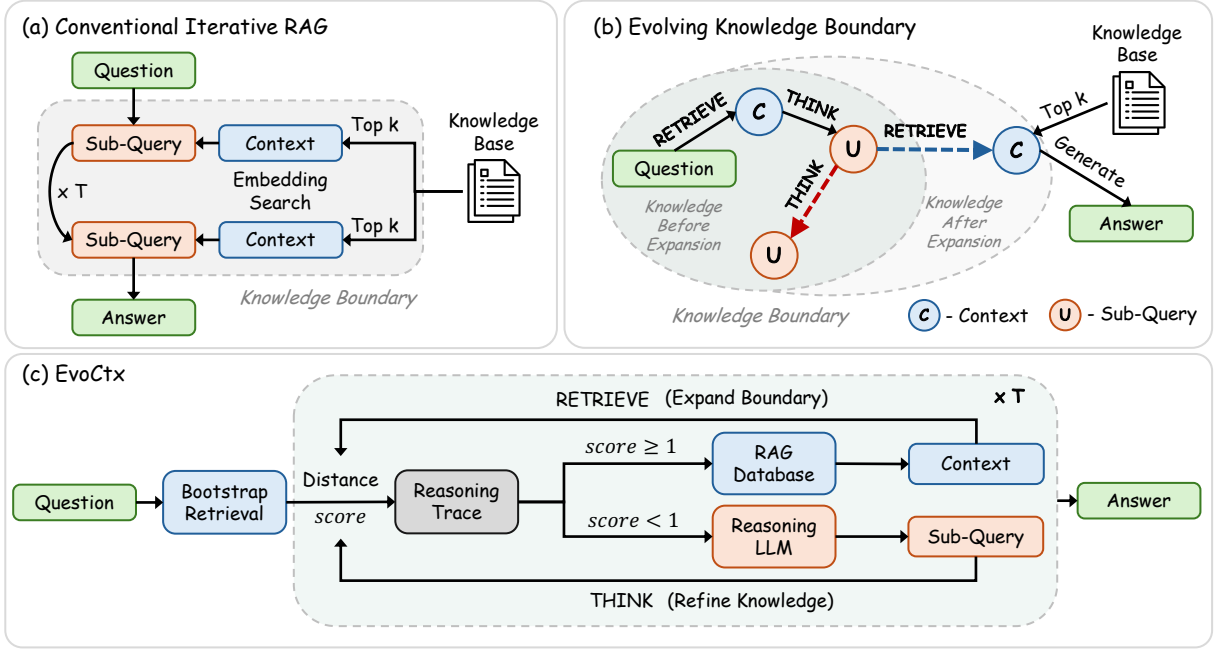


Figure 2: Overview of the proposed **EvoCtx** against previous methods. (a) Conventional iterative RAG retrieves at every step, often causing context saturation. (b) Prior knowledge boundary evolving frames reasoning as *dynamic context acquisition*, simply expanding the knowledge boundary step by step. (c) Our EvoCtx alternates between boundary expansion and trajectory refinement, yielding a compact, adaptive knowledge state to answer the question.

ile for open-world multi-hop reasoning, where the model must operate under incomplete information and progressively uncover the evidence chain. To relax this assumption, recent iterative methods interleave retrieval and generation across multiple rounds (Trivedi et al., 2023; Shao et al., 2023; Yue et al., 2025; Thompson et al., 2025; Verma et al., 2025). Despite their improvements, many approaches still treat retrieval as a default action at each step, expanding the context according to a fixed or nearly fixed schedule rather than deciding whether new external evidence is actually necessary. As shown in Figure 1, a naive RAG pipeline makes a coarse jump from the question to the answer, often leaving intermediate evidence requirements unresolved. Iterative RAG methods (Yue et al., 2025; Fang et al., 2025) improve coverage by retrieving repeatedly, but their indiscriminate expansion scatters intermediate states with weakly relevant evidence and makes the trajectory unnecessarily long. Conversely, allowing an LLM to decide whether retrieval is needed may reduce context size, but can also introduce unsupported transitions when the next hop lies outside the current evidence boundary. This suggests that the core problem is not simply to retrieve more or less, but to decide whether each reasoning transition should *cross* the current boundary of accumulated evidence (Wei

and Yang, 2012a).

We argue that effective context augmentation for multi-hop reasoning should be framed as *boundary-aware context evolution*. Starting from the input question, the model maintains an accumulated context that defines what information is currently supported by external evidence and prior reasoning. When the next reasoning transition requires information beyond this support, the model should retrieve new evidence to expand the boundary. When the transition can be made within the current boundary, the model should instead reason over the existing context, reorganize the accumulated evidence, and refine the unresolved sub-goal (Jiang et al., 2026). This is aligned with human reasoning: effective reasoners do not search indiscriminately, but alternate between seeking new information and making better use of what they already know (Simon, 1983; Ackerman and Thompson, 2017).

Motivated by this perspective, we propose **Cross-Boundary Context Evolution (EvoCtx)**, a framework that turns retrieve-versus-think into a semantic decision over an evolving evidence boundary. EvoCtx first performs a lightweight bootstrap retrieval to ground the question and establish an initial context. It then iteratively selects between two actions: (i) **RETRIEVE**, which expands the current boundary with new external evidence; and (ii)

THINK, which performs intra-boundary reasoning by reorganizing existing evidence, identifying the unresolved semantic gap, and refining the next sub-query without injecting additional contexts. Crucially, EvoCtx estimates the semantic gap between the reasoning-trace embedding and the knowledge accumulated so far. A large gap indicates that the current context no longer supports the next transition and should be expanded through retrieval, whereas a smaller but unresolved gap suggests that the model should mainly refine its context within the current boundary.

Our main contributions are as follows:

- We formulate open-domain multi-hop reasoning as a problem of *boundary-aware context evolution*, shifting the focus from static context construction to adaptive control over how external evidence is gathered.
- We propose **EvoCtx**, a simple yet effective framework that applies this formulation through semantic action selection, estimating the gap between the reasoning embedding and accumulated evidence.
- Extensive experiments on challenging open-domain and multi-hop QA benchmarks show that EvoCtx improves answer accuracy significantly. Further analyses demonstrate that our method yields more compact and evidence-supported reasoning trajectories.

## 2 Related Work

**Retrieval-Augmented Generation.** Retrieval-augmented generation (RAG) improves knowledge-intensive generation by grounding language models in external evidence retrieved from a corpus (Lewis et al., 2020; Yu et al., 2024). The original retrieve-then-generate paradigm remains a strong baseline for open-domain and multi-hop QA, but it assumes that the information need is largely exposed by the initial question (Zhu et al., 2021; Mavi et al., 2024). In multi-hop reasoning, however, missing facts often emerge only after partial reasoning, so a single retrieval step can ground the question but cannot revise the search direction as new evidence needs appear (Lin et al., 2024).

**Adaptive Retrieval for Language Models.** Recent work studies whether retrieval should be invoked selectively rather than unconditionally. Self-RAG uses reflection tokens to let the model retrieve

and critique its own generation (Asai et al., 2024); FLARE retrieves based on predicted uncertain future content (Jiang et al., 2023); AdaptiveRAG routes questions to different retrieval strategies by estimated complexity (Jeong et al., 2024); and DRAGIN triggers retrieval from the model’s evolving generation state (Su et al., 2024). These methods share the intuition that retrieval should be controlled rather than blindly appended. Different from using surface tokens, question-level routing, or token-level uncertainty, our method decides by comparing the model’s reasoning trace with the current knowledge boundary, asking whether the existing boundary still supports progress or must be expanded.

**Multi-hop Reasoning.** Complex QA tasks further highlight the need to interleave retrieval with multi-hop reasoning. IRCOT alternates chain-of-thought reasoning and retrieval so that intermediate reasoning can guide later evidence acquisition (Trivedi et al., 2023); KiRAG and IterDRAG similarly improve multi-hop RAG through iterative retrieval–reasoning processes (Fang et al., 2025; Yue et al., 2025). These methods move beyond one-shot retrieval, but they typically tie retrieval to a procedural loop. We build on the same insight while introducing a stronger decision perspective: open-world RAG should not only evolve context, but decide *how* it should evolve. EvoCtx therefore treats inference as dynamic context acquisition and decision-making over an evolving knowledge boundary, choosing between RETRIEVE to expand that boundary and THINK to refine the decision trajectory.

## 3 Methodology

We propose **EvoCtx**, a framework that formulates retrieval-augmented multi-hop complex reasoning as *dynamic context acquisition and decision making*. Starting from a bootstrap retrieval, EvoCtx iteratively chooses between two actions: RETRIEVE, which expands the current knowledge boundary with new external evidence, and THINK, which refines the current search trajectory by generating a new sub-query under the existing boundary. Figure 2 shows the overview of EvoCtx.

### 3.1 Problem Setup

Given a question  $Q$  and an external corpus  $\mathcal{K}$ , EvoCtx performs one bootstrap retrieval followed by  $T$  adaptive decision rounds. Since  $Q$  and  $\mathcal{K}$

remain fixed throughout inference, we define the mutable state at round  $i$  as:

$$\mathcal{S}_i = (\mathcal{C}_i, \mathcal{U}_i), \quad (1)$$

where  $\mathcal{C}_i$  is the accumulated retrieved context and  $\mathcal{U}_i$  is the set of generated sub-queries. The action space is defined as follows:

$$\mathcal{O} = \{\text{RETRIEVE}, \text{THINK}\}. \quad (2)$$

Then, EvoCtx begins with a bootstrap retrieval using the original question:

$$\mathcal{C}_0 = f_{\text{ret}}(Q, \mathcal{K}), \quad \mathcal{U}_0 = \emptyset. \quad (3)$$

This step grounds the reasoning process in external evidence before adaptive control begins.

### 3.2 Boundary-Aware Decision-Making

At each round  $i$ , a decision model reads the question, current context, and previous sub-queries, and produces a reasoning trace together with an optional surface-form action suggestion:

$$(r_i, \hat{o}_i) = f_{\text{dec}}(Q, \mathcal{C}_i, \mathcal{U}_i). \quad (4)$$

Here  $r_i$  is the reasoning trace, while  $\hat{o}_i \in \mathcal{O}$  is the action token produced by the model. EvoCtx does *not* directly execute  $\hat{o}_i$ . Instead, it uses  $r_i$  as a semantic probe of the model’s current information need.

**Knowledge Boundary.** To decide whether the current boundary is sufficient, we construct a bounded knowledge boundary as follows:

$$\mathcal{B}_i = \psi(Q, \mathcal{U}_i, \mathcal{C}_i), \quad (5)$$

where  $\psi(\cdot)$  collects a deduplicated subset of the evidence accumulated so far, including the original question, previously generated sub-queries, and retrieved passages. In practice,  $\psi(\cdot)$  can bound the number of sub-queries and passages used for decision making, which keeps the boundary representation compact and prevents old or redundant context dominating the decision signal.

**Action Selection with Novelty Score.** Let  $E(\cdot)$  be a text encoder. We encode the reasoning trace and the current boundary as:

$$h_i = E(r_i), \quad H_i = \{E(b) \mid b \in \mathcal{B}_i\}. \quad (6)$$

EvoCtx then measures how far the reasoning embedding moves beyond the current boundary. Let

$\mu_i = |\mathcal{B}_i|^{-1} \sum_{b \in \mathcal{B}_i} E(b)$ . We compute three novelty signals:

$$d_i^{\text{near}} = \min_{b \in \mathcal{B}_i} \delta(h_i, E(b)), \quad (7)$$

$$d_i^{\text{cent}} = \delta(h_i, \mu_i), \quad (8)$$

$$d_i^{90} = Q_{0.9}\{\delta(h_i, E(b))\}_{b \in \mathcal{B}_i}, \quad (9)$$

where  $\delta(x, y) = 1 - \cos(x, y)$ .

These signals capture different aspects of boundary mismatch.  $d_i^{\text{near}}$  measures whether the reasoning embedding is unsupported even by its closest boundary item;  $d_i^{\text{cent}}$  measures whether it drifts away from the global center of the boundary; and  $d_i^{90}$  captures upper-tail mismatch when the reasoning trace is far from most boundary items.

We aggregate these signals into a threshold-normalized novelty score:

$$\text{score} = \max \left\{ \min \left( \frac{d_i^{\text{near}}}{\tau_{\text{near}}}, \frac{d_i^{\text{cent}}}{\tau_{\text{cent}}} \right), \frac{d_i^{90}}{\tau_{90}} \right\}. \quad (10)$$

The executed action is then selected by a compact threshold rule:

$$o_i = \begin{cases} \text{RETRIEVE}, & \text{score} \geq 1, \\ \text{THINK}, & \text{score} < 1. \end{cases} \quad (11)$$

Therefore, if the reasoning trace lies outside the semantic support of the current boundary, the system should retrieve to expand the boundary; otherwise, it should think within the current boundary and refine the search trajectory.

### 3.3 Context Evolution

Once the action  $o_i$  is selected, EvoCtx updates the context accordingly.

**RETRIEVE for Boundary Expansion.** If  $o_i = \text{RETRIEVE}$ , EvoCtx forms a retrieval query by combining the original question with the accumulated sub-queries, given by:

$$q_i^{\text{ret}} = \phi(Q, \mathcal{U}_i), \quad (12)$$

and retrieves new evidence from the corpus as follows:

$$D_i = f_{\text{ret}}(q_i^{\text{ret}}, \mathcal{K} \setminus \mathcal{C}_i), \quad (13)$$

where  $D_i$  denotes the newly retrieved passages and previously seen documents are excluded. The state update is given by:

$$\mathcal{C}_{i+1} = \mathcal{C}_i \cup D_i, \quad \mathcal{U}_{i+1} = \mathcal{U}_i. \quad (14)$$

This action explicitly enlarges the semantic coverage of the current knowledge boundary.

---

**Algorithm 1** Inference Process of EvoCtx

---

**Input:** Question  $Q$ , corpus  $\mathcal{K}$ , adaptive round budget  $T$ , retriever  $f_{\text{ret}}$ , decision model  $f_{\text{dec}}$ , sub-query generator  $f_{\text{sub}}$ , answer generator  $f_{\text{ans}}$ , encoder  $E$ , decision rule  $\text{score}$ .

**Output:** Answer  $f_{\text{ans}}(Q)$ .

```
1:  $\mathcal{U}_0 \leftarrow \emptyset$ 
2:  $\mathcal{C}_0 \leftarrow f_{\text{ret}}(Q, \mathcal{K})$   $\triangleright$  bootstrap retrieval
3: for  $i = 0$  to  $T - 1$  do
4:    $(r_i, \hat{o}_i) \leftarrow f_{\text{dec}}(Q, \mathcal{C}_i, \mathcal{U}_i)$ 
5:    $\mathcal{B}_i \leftarrow \psi(Q, \mathcal{U}_i, \mathcal{C}_i)$ 
6:   Compute novelty signals  $(d_i^{\text{near}}, d_i^{\text{cent}}, d_i^{90})$ 
     from  $E(r_i)$  and  $\{E(b) : b \in \mathcal{B}_i\}$ 
7:    $o_i \leftarrow \text{score}(d_i^{\text{near}}, d_i^{\text{cent}}, d_i^{90})$ 
8:   if  $o_i = \text{RETRIEVE}$  then
9:      $q_i^{\text{ret}} \leftarrow \phi(Q, \mathcal{U}_i)$ 
10:     $D_i \leftarrow f_{\text{ret}}(q_i^{\text{ret}}, \mathcal{K} \setminus \mathcal{C}_i)$ 
11:     $\mathcal{C}_{i+1} \leftarrow \mathcal{C}_i \cup D_i, \mathcal{U}_{i+1} \leftarrow \mathcal{U}_i$ 
12:   else
13:      $u_i \leftarrow f_{\text{sub}}(Q, \mathcal{C}_i, \mathcal{U}_i)$ 
14:      $\mathcal{U}_{i+1} \leftarrow \mathcal{U}_i \cup \{u_i\}, \mathcal{C}_{i+1} \leftarrow \mathcal{C}_i$ 
15:   end if
16: end for
17: return  $f_{\text{ans}}(Q, \mathcal{C}_T, \mathcal{U}_T)$ 
```

---

**THINK for Trajectory Refinement.** If  $o_i = \text{THINK}$ , a generation model proposes one new sub-query conditioned on the current state:

$$u_i = f_{\text{sub}}(Q, \mathcal{C}_i, \mathcal{U}_i). \quad (15)$$

The state is then updated as:

$$\mathcal{U}_{i+1} = \mathcal{U}_i \cup \{u_i\}, \quad \mathcal{C}_{i+1} = \mathcal{C}_i. \quad (16)$$

Importantly, THINK does not inject new external evidence or produce an intermediate answer. Its core role is to **refine the query space under the current evidence boundary**: it reorganizes the next search direction from the currently supported context so that later retrieval, if needed, is better targeted and more answer-relevant. Equivalently, RETRIEVE updates only  $\mathcal{C}$ , while THINK updates only  $\mathcal{U}$ .

### 3.4 Final Answer Generation

After  $T$  adaptive rounds, the final answer is generated from the question, the accumulated context, and the generated sub-queries:

$$A = f_{\text{ans}}(Q, \mathcal{C}_T, \mathcal{U}_T). \quad (17)$$

The final prediction is therefore conditioned on a dynamically constructed knowledge state rather

than on a single static retrieval result. Algorithm 1 summarizes the full EvoCtx inference process.

## 4 Experiments

### 4.1 Experimental Setup

**Benchmarks.** We evaluate on four QA benchmarks that cover both multi-hop and open-domain evidence-seeking behavior. 2WikiMulti-hopQA (Ho et al., 2020) and HotpotQA (Yang et al., 2018) require combining evidence across entities or documents. MultiHop-RAG (Tang and Yang, 2024) further stresses retrieval-augmented multi-hop reasoning in settings where the relevant evidence is distributed across retrieved passages. TriviaQA (Joshi et al., 2017) provides an additional open-domain QA setting for testing whether a method can acquire useful external evidence beyond multi-hop-only datasets.

**Implementation Details.** EvoCtx decouples action selection from answer generation. We use Qwen3-4B-Thinking (Team, 2025) as the decision model that uses its thinking contents’ semantics to determine whether the next step should be RETRIEVE or THINK, and LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024) as the generation model for sub-query generation and final answer prediction. Semantic comparison between the reasoning trace embedding and the current knowledge boundary is implemented with the all-MiniLM-L6-v2 text encoder. The parameter setting has two major variables: the total rounds of the iteration ( $T$ ) and the number of chunks to be retrieved ( $k$ ) is set to be 8 and 5. The boundary thresholds are set as:  $\tau_{\text{near}} = 0.3$ ,  $\tau_{\text{cent}} = 0.2$ , and  $\tau_{90} = 0.4$ . In the main experiments, we set the adaptive round budget to  $T = 8$  and the retrieval depth to  $\text{top-}k = 5$ .

**Baseline Methods.** We compare against the following methods with different decision types: **Closed-Book** baselines ask LLaMA-3.1-8B-Instruct and Qwen3-4B-Instruct to answer directly from internal knowledge. **Static** retrieval is represented by NaiveRAG (Lewis et al., 2020), which performs one retrieval step and then answers from a fixed context. **Iterative** methods, including IR-CoT (Trivedi et al., 2023), KiRAG (Fang et al., 2025), and IterDRAG (Yue et al., 2025), repeatedly couple retrieval with intermediate reasoning, but their decision process is largely tied to a predefined loop. **Adaptive** methods, including Adap-

| Method                             | Decision Type        | 2Wiki |      |      | HotPotQA |      |      | MultiHop-RAG |      |      | TriviaQA |      |      |
|------------------------------------|----------------------|-------|------|------|----------|------|------|--------------|------|------|----------|------|------|
|                                    |                      | EM    | F1   | Acc  | EM       | F1   | Acc  | EM           | F1   | Acc  | EM       | F1   | Acc  |
| Internal Knowledge Only            |                      |       |      |      |          |      |      |              |      |      |          |      |      |
| Llama-3.1-8B-Instruct              | Closed-Book          | 0.0   | 5.4  | 48.4 | 0.0      | 4.4  | 49.2 | 7.0          | 7.9  | 37.6 | 0.0      | 8.6  | 69.7 |
| Qwen3-4B-Instruct                  | Closed-Book          | 0.0   | 2.5  | 51.8 | 0.0      | 1.9  | 50.5 | 6.2          | 6.4  | 39.6 | 0.1      | 3.3  | 56.3 |
| Retrieval-based External Knowledge |                      |       |      |      |          |      |      |              |      |      |          |      |      |
| NaiveRAG (Lewis et al., 2020)      | Static               | 30.0  | 38.2 | 48.6 | 38.2     | 51.9 | 56.8 | 36.7         | 39.2 | 45.6 | 53.6     | 62.4 | 60.2 |
| IRCoT (Trivedi et al., 2023)       | Iterative            | 36.2  | 46.0 | 45.4 | 41.4     | 51.6 | 43.8 | —            | —    | —    | 52.6     | 63.4 | 60.8 |
| KiRAG (Fang et al., 2025)          | Iterative            | 30.7  | 50.6 | —    | 45.1     | 59.8 | —    | —            | —    | —    | —        | —    | —    |
| IterDRAG (Yue et al., 2025)        | Iterative            | 31.8  | 38.4 | 47.7 | 34.6     | 44.8 | 51.8 | 59.8         | 60.8 | 58.4 | 16.9     | 21.9 | 60.5 |
| AdaptiveRAG (Jeong et al., 2024)   | Adaptive             | 38.4  | 47.1 | 45.4 | 39.6     | 49.9 | 41.4 | —            | —    | —    | 56.4     | 65.6 | 62.8 |
| Self-RAG (Asai et al., 2024)       | Adaptive             | 12.3  | 25.1 | 33.3 | 17.4     | 29.9 | 32.1 | —            | —    | —    | 12.8     | 29.3 | 57.0 |
| FLARE (Jiang et al., 2023)         | Adaptive             | 35.8  | 45.1 | 42.4 | 29.8     | 39.1 | 37.2 | —            | —    | —    | 57.0     | 67.4 | 64.8 |
| DRAGIN (Su et al., 2024)           | Adaptive             | 40.6  | 48.0 | 45.6 | 39.8     | 50.6 | 43.0 | —            | —    | —    | 58.4     | 69.1 | 66.6 |
| EvoCtx (Ours)                      | Adaptive + Iterative | 46.2  | 50.7 | 53.6 | 49.1     | 63.0 | 65.4 | 74.6         | 74.6 | 66.3 | 64.3     | 72.3 | 69.8 |

Table 1: Main results comparing EvoCtx with baseline methods on complex reasoning benchmarks. We utilize 2Wiki, HotpotQA, and MultiHop-RAG to measure multi-hop reasoning capabilities, while including TriviaQA as a complementary single-hop knowledge-intensive task. Best results are highlighted in **bold**.

tiveRAG (Jeong et al., 2024), Self-RAG (Asai et al., 2024), FLARE (Jiang et al., 2023), and DRAGIN (Su et al., 2024), decide retrieval more flexibly using complexity, reflection, uncertainty, or information-need signals. As an **Adaptive + Iterative** method, our EvoCtx uses reasoning trace embedding to judge whether the current knowledge boundary is sufficient, thereby deciding between THINK and RETRIEVE. The baseline settings are aligned with EvoCtx where  $T = 8$  and  $k = 5$ .

For retrieval-based comparisons, we align shared components whenever the baseline interface is plug-compatible. In particular, we use the same benchmark corpus and final answer generator (LLaMA-3.1-8B-Instruct), and matched upper bounds on the adaptive budget and retrieval depth where these controls are exposed. When a baseline requires its own prompt format, controller, or training recipe by design, we preserve the original implementation rather than forcing artificial standardization. This protocol is intended to keep shared components aligned where meaningful while maintaining the native behavior of each baseline.

**Evaluation Metrics.** Following many existing studies (Lewis et al., 2020; Yue et al., 2025), we adopt Exact Match (EM), token-level F1 (F1), and Accuracy (Acc) as the evaluation metrics. EM and F1 measure answer overlap under standard normalization, while Acc measures whether the prediction contains an acceptable ground-truth answer.

## 4.2 Main Results

Table 1 summarizes the main comparison. Overall, the results support a decision-centric view of RAG: performance improves as methods move

from Closed-Book internal knowledge to Static, Iterative, and Adaptive external-knowledge use, but the strongest results require a cognitive decision over whether to RETRIEVE or THINK. EvoCtx achieves the best scores across all four benchmarks, with especially large gains on MultiHop-RAG.

**External knowledge must be crossed, but not only once.** Closed-Book baselines obtain near-zero EM on 2Wiki and HotPotQA, showing that internal parametric knowledge is insufficient for exact, evidence-grounded reasoning. NaiveRAG improves substantially by crossing the boundary to external evidence, but the gap between NaiveRAG and EvoCtx shows that one static retrieval step cannot handle information needs that emerge after partial reasoning. Iterative methods such as IRCoT (Trivedi et al., 2023), KiRAG (Fang et al., 2025), and IterDRAG (Yue et al., 2025) address this by repeatedly coupling retrieval with reasoning, yet their retrieval process is still largely procedural. EvoCtx improves by making each step conditional on whether the current knowledge boundary semantically supports the reasoning trace embedding.

**Boundary-aware control improves over adaptive triggering.** Adaptive methods use complexity, uncertainty, reflection, or information-need signals to decide retrieval and are strong in several settings, but their performance varies across datasets, suggesting that these signals do not always match the true evidence need. EvoCtx advances this adaptive philosophy into a boundary-aware cognitive decision type: it checks whether the current reasoning state is already supported by the accumulated boundary, using THINK to refine the trajectory when

| Variant              | EM          | F1          | Acc         |
|----------------------|-------------|-------------|-------------|
| <b>EvoCtx (Ours)</b> | <b>74.6</b> | <b>74.6</b> | <b>66.3</b> |
| w/o RETRIEVE         | 62.9        | 64.2        | 58.4        |
| w/o THINK            | 71.1        | 71.6        | 64.0        |
| w/o Reasoning        | 63.7        | 64.1        | 58.4        |
| w/o Boundary         | 68.4        | 68.8        | 61.8        |

Table 2: Ablation study on the MultiHop-RAG benchmark. Each variant removes one component of EvoCtx’s dynamic retrieve-or-think decision process.

| $T$  | Top- $k$ Retrieved Passages |      |      |      |             | Avg.        |
|------|-----------------------------|------|------|------|-------------|-------------|
|      | 1                           | 2    | 3    | 4    | 5           |             |
| 2    | 60.0                        | 55.0 | 60.0 | 60.0 | 64.0        | 59.8        |
| 3    | 61.0                        | 57.0 | 66.0 | 64.0 | 67.0        | 63.0        |
| 4    | 59.0                        | 58.0 | 59.0 | 68.0 | 63.0        | 61.4        |
| 5    | 56.0                        | 59.0 | 58.0 | 63.0 | 65.0        | 60.2        |
| 6    | 55.0                        | 65.0 | 64.0 | 63.0 | 68.0        | 63.0        |
| 7    | 60.0                        | 56.0 | 60.0 | 62.0 | 71.0        | 61.8        |
| 8    | 60.0                        | 65.0 | 66.0 | 68.0 | <b>72.0</b> | <b>66.2</b> |
| Avg. | 58.7                        | 59.3 | 61.9 | 64.0 | <b>67.1</b> | –           |

Table 3: Sensitivity of EvoCtx to the adaptive round budget  $T$  and the number of retrieved passages  $k$  on a 100-example MultiHop-RAG subset. Each value denotes EM (%). The best setting is highlighted in bold.

supported and RETRIEVE to expand the boundary when unsupported. The largest gain on MultiHop-RAG, where EvoCtx reaches 74.6 EM/F1 compared with 59.8/60.8 for IterDRAG, directly supports our central claim that better RAG is not simply more retrieval, but high-level control over when to remain inside the boundary and when to cross it.

### 4.3 Ablation Study

Table 2 shows that each part of EvoCtx contributes to the final performance. **w/o RETRIEVE** disables evidence acquisition after the bootstrap step, so the model can only continue refining its trajectory using THINK. Its large drop confirms that bootstrap evidence alone cannot satisfy later-emerging information needs in open-world multi-hop reasoning. **w/o THINK** removes intermediate trajectory refinement and forces every subsequent step to retrieve new evidence, namely to act RETRIEVE. The resulting degradation shows that blind boundary expansion is less effective than selectively reasoning within the current boundary before retrieving again. **w/o Reasoning** replaces the reasoning-based decider with LLaMA-3.1-8B and treats the control decision as a shallow action prediction without explicit reasoning trace. Its weaker performance indicates that the model must expose its current information need through reasoning in order to decide whether

| Hybrid / Tail-Based Rules |               |               |             |             | Single-Signal Rules |               |               |             |      |
|---------------------------|---------------|---------------|-------------|-------------|---------------------|---------------|---------------|-------------|------|
| Rule                      | $\tau_{near}$ | $\tau_{cent}$ | $\tau_{90}$ | EM          | Rule                | $\tau_{near}$ | $\tau_{cent}$ | $\tau_{90}$ | EM   |
| hybrid                    | 0.3           | 0.2           | 0.4         | <b>72.0</b> | nearest             | 0.3           | –             | –           | 63.0 |
| hybrid                    | 0.4           | 0.3           | 0.5         | 64.0        | nearest             | 0.4           | –             | –           | 66.0 |
| hybrid                    | 0.5           | 0.4           | 0.6         | 68.0        | nearest             | 0.5           | –             | –           | 62.0 |
| distribution              | –             | –             | 0.4         | 68.0        | centroid            | –             | 0.2           | –           | 67.0 |
| distribution              | –             | –             | 0.5         | 64.0        | centroid            | –             | 0.3           | –           | 58.0 |
| distribution              | –             | –             | 0.6         | 54.0        | centroid            | –             | 0.4           | –           | 62.0 |

Table 4: Threshold sensitivity on the MultiHop-RAG test subset (100 examples,  $N=8$ ,  $K=5$ ). “–” denotes that the entries are unused.

the boundary is sufficient. **w/o Boundary** still uses the reasoning LLM, but directly executes its generated action token without checking whether the reasoning embedding is semantically supported by the accumulated boundary. The drop relative to full EvoCtx demonstrates that semantic verification is necessary to turn reasoning into a reliable boundary-control signal.

**Sensitivity to the round budget  $T$  and retrieval depth  $k$ .** Table 3 examines how EvoCtx behaves under different adaptive round budgets and retrieval depths. Two patterns are clear. First, larger  $k$  is consistently beneficial: the average EM rises from 58.7 at  $k = 1$  to 67.1 at  $k = 5$ , indicating that when the controller decides to cross the current knowledge boundary, expanding it with sufficiently rich evidence is important. Second, increasing  $T$  is most helpful when paired with adequate retrieval capacity, with the best result achieved at  $T = 8$  and  $k = 5$ . At the same time, the gains are not strictly monotonic for every setting, which supports our central claim: performance does not come from blindly adding more iterations or more retrieved passages, but from enabling effective cognitive control over THINK-versus-RETRIEVE decisions as the knowledge boundary evolves.

**Threshold sensitivity.** Table 4 examines how different threshold rules instantiate EvoCtx’s semantic boundary on a 100-example MultiHop-RAG subset. The hybrid rule performs best, reaching 72 EM under permissive thresholds, suggesting that EvoCtx benefits from retrieving as soon as multiple novelty signals jointly indicate that the current frontier is insufficient. By contrast, single-signal rules are weaker and less stable: nearest and centroid never match the hybrid peak, while the distribution-only rule degrades as  $\tau_{90}$  becomes more conservative. Overall, this supports our design: cognitive control is more reliable when novelty is judged from complementary geometric signals.

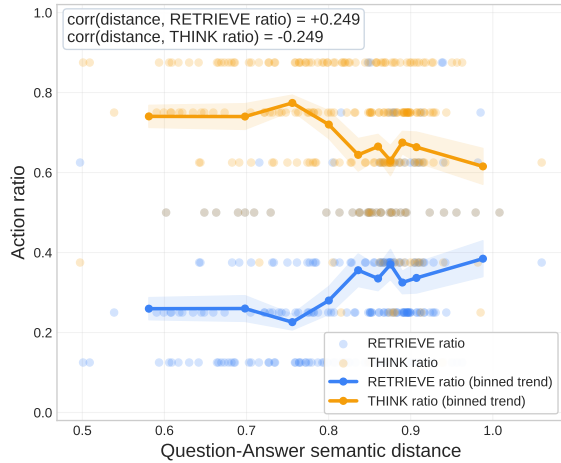


Figure 3: Action allocation with respect to question-answer semantic distance. Each point reports the ratio of RETRIEVE or THINK actions for one example, while solid curves show binned trends. Larger semantic distance is associated with a higher retrieval ratio and a lower thinking ratio, suggesting that EvoCtx adaptively expands the knowledge boundary more often when the answer is semantically farther from the question.

#### 4.4 In-depth Analysis

##### Action allocation follows semantic difficulty.

Figure 3 analyzes whether EvoCtx’s decisions reflect example-level semantic difficulty rather than following a fixed retrieval schedule. We use the question-answer semantic distance as a proxy for how far the reasoning process must move from the initial question to the target ground truth. The positive correlation between distance and RETRIEVE ratio, together with the symmetric negative correlation for THINK, shows that EvoCtx tends to retrieve more when the answer lies farther beyond the initial semantic neighborhood. This supports our boundary-crossing interpretation, where harder cases require more frequent expansion of the knowledge boundary. Importantly, the trend is moderate rather than absolute; even high-distance examples retain substantial THINK actions. This indicates that EvoCtx does not reduce difficulty to simply retrieving more documents. Instead, it coordinates retrieval with trajectory refinement: RETRIEVE expands the boundary when evidence is missing, while THINK keeps the search direction coherent so that later retrieval is better targeted.

**Inference-time decisions align with oracle evidence needs.** Figure 4 evaluates whether EvoCtx’s inference-time decision signal is a faithful proxy for the true need for external evidence.

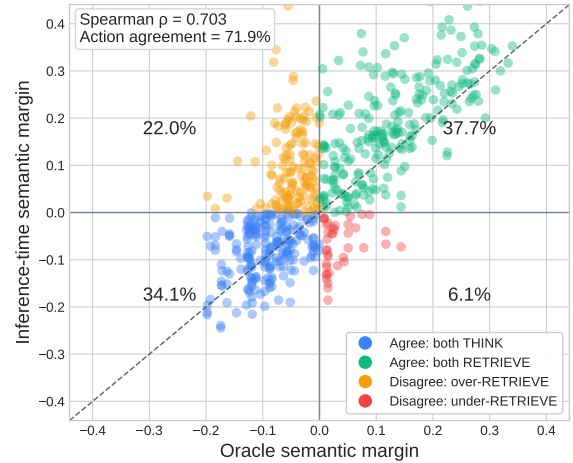


Figure 4: Inference-time versus post hoc oracle semantic margin at each EvoCtx decision round. Each point is one round under the same knowledge boundary. Positive margins indicate RETRIEVE and negative margins indicate THINK. The strong correlation and 71.9% action agreement show that EvoCtx’s reasoning signal usually matches the oracle decision on whether to expand the boundary or continue reasoning within it.

For each decision round, we compare the signed semantic margin ( $score - 1$ ) computed from the model’s reasoning trace with a post hoc oracle semantic margin computed from ground-truth target semantics under the same knowledge boundary. The strong positive correlation ( $\rho = 0.703$ ) shows that when the reasoning state moves farther beyond the current boundary, the oracle assessment of missing information also tends to increase. Moreover, 71.9% of decision points fall in the agreement quadrants, indicating that EvoCtx usually makes the same RETRIEVE-versus-THINK choice as the oracle. The disagreement pattern is also asymmetric: the more concerning under-RETRIEVE region, where the oracle indicates that the boundary should be expanded but EvoCtx remains in THINK, accounts for only 6.1% of cases, while most disagreement comes from over-RETRIEVE decisions. Despite over-RETRIEVE being able to somehow incorporate noisy data into the knowledge, the robustness for the question-answering task is stable due to enough information gain. This pattern is consistent with our main claim that effective open-world RAG is not simply repeated retrieval, but cognitive control over when to cross the current knowledge boundary and when to continue refining the reasoning trajectory.

**Efficiency and practical tradeoffs.** EvoCtx is designed to improve retrieval allocation rather than

| Method        | Avg. retrieval calls | Avg. retrieved passages | Avg. tokens consumed |
|---------------|----------------------|-------------------------|----------------------|
| NaiveRAG      | 1.0                  | 5.0                     | 1.1k                 |
| IterDRAG      | 8.0                  | 17.6                    | 18.2k                |
| EvoCtx (Ours) | 3.2                  | 12.8                    | 27.5k                |

Table 5: Inference-time efficiency on MultiHop-RAG. EvoCtx uses fewer retrieval calls and fewer retrieved passages than fixed iterative retrieval, while consuming more total tokens due to reasoning traces and THINK steps.

to minimize latency at all costs. Table 5 shows that, on MultiHop-RAG, EvoCtx uses substantially fewer retrieval calls than fixed iterative retrieval and retrieves fewer total passages, indicating a more selective expansion of the evidence boundary. At the same time, EvoCtx can consume more total tokens because THINK steps expose reasoning traces and refine sub-queries before later retrieval. This tradeoff is consistent with our goal: EvoCtx is retrieval-efficient relative to fixed iterative retrieval, while remaining more computation-heavy than one-shot static RAG.

## 5 Conclusion

We presented EvoCtx, a framework that formulates retrieval-augmented multi-hop reasoning as *dynamic context acquisition and decision making* over an evolving knowledge boundary. Instead of treating retrieval as the default action, EvoCtx decides whether to RETRIEVE or THINK by measuring the semantic gap between the model’s reasoning trace and the knowledge accumulated so far. Experiments on challenging multi-hop and open-domain QA benchmarks show that this boundary-aware retrieve-versus-think control improves performance over static, iterative, and adaptive baselines. Our results support a claim that better open-world RAG is not only about retrieving more evidence, but about deciding when the current boundary should be expanded and when reasoning should continue within it.

## Limitations

Our evaluation is conducted on publicly available question answering benchmarks and under a standard text-retrieval setting, which together provide a relatively clean, controlled, and reproducible testbed for studying metacognitive retrieve-versus-think decisions. This setup allows us to examine the proposed mechanism without introducing additional confounding factors from more specialized environments or heterogeneous retrieval conditions. While the selected datasets already cover both

multi-hop reasoning and open-domain evidence-seeking behavior, they still represent only a limited portion of the broader landscape of knowledge-intensive tasks. Therefore, further validation on a wider range of domains would help strengthen the evidence for the framework’s robustness and improve confidence in its generality across more diverse application scenarios.

## Ethics Statement

This work is intended to improve the reliability of retrieval-augmented reasoning. Our experiments use public benchmarks and standard open-sourced language models, and we do not collect human subject data. Like other retrieval-augmented systems, the method may inherit inaccuracies, omissions, or social biases from the underlying language models and external corpora. However, by explicitly deciding when new evidence is needed and by maintaining an evidence-supported reasoning trajectory, EvoCtx can provide a more transparent basis for knowledge acquisition than fixed retrieval loops. For real-world deployment, standard safeguards such as corpus curation, source tracking, and human oversight remain important, especially in sensitive application settings.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. T2541073 and No. 62372314. The experimental part of this work was supported by The Centre for Large AI Models (CLAIM) of The Hong Kong Polytechnic University.

## References

- Rakefet Ackerman and Valerie A Thompson. 2017. Meta-reasoning: Monitoring and control of thinking and reasoning. *Trends in cognitive sciences*, 21(8):607–617.
- Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection.

- In *International conference on learning representations*, volume 2024, pages 9112–9141.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.
- 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.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. *The llama 3 herd of models*. Preprint, arXiv:2407.21783.
- 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.
- Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. 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 7036–7050.
- Yiyang Jiang, Li Zhang, Xiao-Yong Wei, and Qing Li. 2026. Think in latent thoughts: A new paradigm for gloss-free sign language translation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9993–10012.
- 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.
- Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.
- 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.
- Zefeng Lin, Weidong Chen, Yan Song, and Yongdong Zhang. 2024. Prompting few-shot multi-hop question generation via comprehending type-aware semantics. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3730–3740.
- Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2024. Multi-hop question answering. *Foundations and Trends® in Information Retrieval*, 17(4):457–586.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. 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.
- Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274.
- Herbert A. Simon. 1983. Search and reasoning in problem solving. *Artif. Intell.:(Netherlands)*, 1.
- Weihsang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. Dragin: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12991–13013.
- Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*.
- Qwen Team. 2025. *Qwen3 technical report*. Preprint, arXiv:2505.09388.
- Travis Thompson, Seung-Hwan Lim, Paul Liu, Ruoying He, and Dongkuan Xu. 2025. Inference scaled graphrag: Improving multi hop question answering on knowledge graphs. *arXiv preprint arXiv:2506.19967*.
- 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.

- Prakhar Verma, Sukruta Prakash Midigeshi, Gaurav Sinha, Arno Solin, Nagarajan Natarajan, and Amit Sharma. 2025. Plan\*rag: Efficient test-time planning for retrieval augmented generation. In *Workshop on Reasoning and Planning for Large Language Models*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Xiao-Yong Wei and Zhen-Qun Yang. 2012a. Coaching the exploration and exploitation in active learning for interactive video retrieval. *IEEE transactions on image processing*, 22(3):955–968.
- Xiao-Yong Wei and Zhen-Qun Yang. 2012b. Mining in-class social networks for large-scale pedagogical analysis. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 639–648.
- 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 *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. 2024. Evaluation of retrieval-augmented generation: A survey. In *CCF Conference on Big Data*, pages 102–120. Springer.
- Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xu-anhui Wang, and Michael Bendersky. 2025. Inference scaling for long-context retrieval augmented generation. In *International Conference on Learning Representations*, volume 2025, pages 72914–72938.
- Wengyu Zhang, Xiao-Yong Wei, and Qing Li. 2026. Mol-debate: multi-agent debate improves structural reasoning in molecular design. *arXiv preprint arXiv:2604.20254*.
- Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhen-gren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. 2026. Retrieval-augmented generation for ai-generated content: A survey. *Data Science and Engineering*, 11(1):1–29.
- Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.

## A Additional Robustness Analyses

This appendix reports additional robustness results that are omitted from the main paper due to space constraints. These experiments are designed to test whether EvoCtx’s gain depends heavily on a particular threshold choice, signal combination, or representation backend. Overall, the appendix supports the same conclusion as the main paper: EvoCtx is not merely benefiting from one lucky configuration, but from a stable boundary-aware decision mechanism.

### A.1 Cross-dataset Threshold Robustness

To examine whether the controller is overly sensitive to the exact threshold configuration, we compare several rule families on two datasets and report the best setting within each family. Table 6 is intended to answer two questions: (1) whether the full hybrid controller is consistently strong across datasets, and (2) whether simplified rules can recover most of the gain. The results show that the full hybrid rule remains the strongest on both MultiHop-RAG and HotPotQA, while simpler variants remain competitive but consistently weaker. This supports our claim that the three signals provide complementary evidence for boundary-aware control, rather than the performance being driven by a single threshold heuristic.

| Dataset      | Rule family / best setting | Best EM |
|--------------|----------------------------|---------|
| MultiHop-RAG | Hybrid (0.3, 0.2, 0.4)     | 72.0    |
| MultiHop-RAG | Near + Tail (0.5, 0.4)     | 71.0    |
| MultiHop-RAG | Near + Centroid (0.3, 0.2) | 70.0    |
| MultiHop-RAG | Centroid + Tail (0.2, 0.5) | 66.0    |
| MultiHop-RAG | Nearest-only (0.4)         | 66.0    |
| MultiHop-RAG | Centroid-only (0.2)        | 67.0    |
| MultiHop-RAG | Tail-only (0.4)            | 68.0    |
| HotPotQA     | Hybrid (0.3, 0.3, 0.4)     | 49.0    |
| HotPotQA     | Near + Tail (0.3, 0.4)     | 48.0    |
| HotPotQA     | Near + Centroid (0.4, 0.3) | 44.0    |
| HotPotQA     | Centroid + Tail (0.3, 0.5) | 46.0    |
| HotPotQA     | Nearest-only (0.5)         | 45.0    |
| HotPotQA     | Centroid-only (0.3)        | 39.0    |
| HotPotQA     | Tail-only (0.4)            | 41.0    |

Table 6: Best threshold settings across datasets and rule families. The full hybrid rule remains strongest, while simplified variants remain competitive.

### A.2 Pairwise Signal Ablations

We further isolate the contribution of each signal by retaining only pairwise combinations of the three novelty signals. Table 7 complements Table 6 by testing whether all three signals are necessary, or

whether two are already sufficient. The strongest pairwise controller is *Nearest + Tail*, suggesting that local support and broad mismatch together capture most of the useful decision signal. However, the full hybrid controller still performs best overall, which supports our design choice of using all three signals for the most reliable boundary judgment.

| Signals kept       | Best setting    | EM   |
|--------------------|-----------------|------|
| Nearest + Centroid | (0.3, 0.2)      | 70.0 |
| Nearest + Tail     | (0.5, 0.4)      | 71.0 |
| Centroid + Tail    | (0.2, 0.5)      | 66.0 |
| Full Hybrid        | (0.3, 0.2, 0.4) | 72.0 |

Table 7: Pairwise signal ablations on MultiHop-RAG. The nearest-plus-tail variant is the strongest pairwise controller, but the full three-signal rule remains best.

### A.3 Reasoning-Trace Model and Boundary Encoder Attribution

The main paper uses QWEN3-4B-THINKING to produce the reasoning trace and ALL-MINILM-L6-V2 to encode the current boundary. To test whether EvoCtx is overly tied to this exact combination, Table 9 varies both the reasoning-trace model and the boundary encoder while keeping the final answer generator fixed. The table is intended to separate two sources of gain: the quality of the reasoning trace and the quality of the boundary representation. The best performance is achieved by the original configuration, indicating that explicit reasoning traces and a lightweight sentence encoder are a good match for our controller. At the same time, the overall trend remains stable across alternative combinations, suggesting that the benefit of EvoCtx comes from the boundary-aware control mechanism itself rather than from a single fragile representation choice.

| Adaptive round $T$ | Avg. confidence score | Delta |
|--------------------|-----------------------|-------|
| 2                  | 0.49                  | —     |
| 3                  | 0.56                  | +0.07 |
| 4                  | 0.72                  | +0.16 |
| 5                  | 0.70                  | -0.02 |
| 6                  | 0.78                  | +0.08 |
| 7                  | 0.81                  | +0.03 |
| 8                  | 0.82                  | +0.01 |

Table 8: Confidence probing on HotPotQA. The score is the logit probability of “Yes” when the model is asked whether the current evidence is sufficient for answering the question. Overall, the evolving context becomes increasingly answerable across steps.

## B Additional Experimental Details

This section provides implementation details that are summarized only briefly in the main paper.

### B.1 Full Implementation Details

Table 10 lists the main implementation choices of EvoCtx, including retrieval backend, chunking strategy, indexing, prompt roles, and normalization. It specifies the exact inference-time setup under which the reported numbers are obtained and highlights that EvoCtx is a training-free controller built on top of standard retriever and generator components.

### B.2 Baseline Alignment Policy

A common concern in RAG comparisons is whether the observed gain comes from the controller itself or from differences in retrievers, prompts, and answer backbones. Table 11 clarifies our alignment policy for baseline comparisons. The table is intended to document which components are standardized whenever possible and which are left unchanged to preserve the native behavior of the compared method. This protocol supports the claim that our comparison is controlled where meaningful, while avoiding artificial modifications that would disadvantage baseline methods.

## C Additional Evidence for the Decision Signal

Beyond end-task accuracy, we also provide supporting evidence that EvoCtx’s controller tracks whether the evolving context is becoming more answer-ready over time.

### C.1 Confidence Probing Across Reasoning Steps

To measure whether the accumulated context becomes increasingly sufficient for answering the question, we probe the model at different adaptive rounds with a binary sufficiency question and record the confidence assigned to the positive response. Table 8 shows that the average confidence generally rises as more retrieve-or-think steps are performed. This supports the claim that EvoCtx is not only collecting more information, but is progressively organizing the context into a more answerable knowledge state. The small local fluctuation is expected because some intermediate steps prioritize sub-query refinement over immediate answerability, but the overall trend remains upward.

| Reasoning trace model | Boundary encoder    | EM   | F1   | Acc  |
|-----------------------|---------------------|------|------|------|
| Qwen3-4B-Thinking     | all-MiniLM-L6-v2    | 75.0 | 75.0 | 66.0 |
| Qwen3-4B-Instruct     | all-MiniLM-L6-v2    | 68.0 | 61.0 | 65.0 |
| LLaMA-3.1-8B-Instruct | all-MiniLM-L6-v2    | 66.0 | 70.0 | 69.0 |
| Qwen3-4B-Thinking     | LLaMA hidden states | 71.0 | 70.0 | 67.0 |
| Qwen3-4B-Instruct     | LLaMA hidden states | 63.0 | 65.0 | 61.0 |
| LLaMA-3.1-8B-Instruct | LLaMA hidden states | 60.0 | 74.0 | 68.0 |

Table 9: Model attribution on MultiHop-RAG with the final answer generator fixed to LLaMA-3.1-8B-Instruct. The overall trend is preserved across reasoning-trace models and boundary encoders.

| Component            | Setting                                                                                                                                                                                                            |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Retriever            | Dense retriever based on ALL-MINILM-L6-v2, using cosine / inner-product similarity over a FAISS flat index.                                                                                                        |
| Corpus and chunking  | For 2WikiMultiHopQA and HotPotQA, we follow the benchmark-provided corpus processing. For MultiHop-RAG and TriviaQA, we use sentence-aware chunking with SentenceSplitter, 256-token chunks, and 32-token overlap. |
| Indexing             | Passage embeddings are precomputed once, normalized, and indexed with flat indexing.                                                                                                                               |
| Retrieval depth      | Bootstrap retrieval and each RETRIEVE step use top- $k = 5$ passages in the main experiments.                                                                                                                      |
| Top- $k$ handling    | Duplicate passages are removed; previously seen documents are excluded; retrieved passages are appended in retrieval-score order and truncated to a 1024-token context budget.                                     |
| Decision prompt      | The prompt asks the decision model to expose its reasoning trace over the current question, accumulated context, and generated sub-queries.                                                                        |
| Sub-query prompt     | The prompt asks the generator to produce one unresolved next-hop sub-query under the current evidence boundary.                                                                                                    |
| Final answer prompt  | The prompt asks the answer model to produce a short-form answer from the final question-context-query state.                                                                                                       |
| Answer normalization | Standard EM/F1 normalization: lowercasing, punctuation/article removal, whitespace normalization, plus dataset-specific aliases where applicable.                                                                  |
| Training             | None. EvoCtx is inference-time only.                                                                                                                                                                               |

Table 10: Full implementation details of EvoCtx.

| Aspect                                             | Treatment                                                                                                                                                                                                                     |
|----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Corpus                                             | The same benchmark corpus is used for all methods.                                                                                                                                                                            |
| Final answer model                                 | We use the same final answer generator (LLaMA-3.1-8B-INSTRUCT) for retrieval-based comparisons whenever the baseline interface is plug-compatible; exceptions are preserved only when required by the original method design. |
| Retriever / index                                  | The same retriever backend is used when the baseline is plug-compatible; otherwise, we preserve the original implementation.                                                                                                  |
| Adaptive budget / retrieval depth / context budget | We match the same upper budget whenever the baseline exposes these knobs.                                                                                                                                                     |
| Prompt style                                       | Baseline-specific when required by the original method design.                                                                                                                                                                |
| Training                                           | We preserve the original baseline training setup; EvoCtx itself is training-free.                                                                                                                                             |
| Controller parameters                              | EvoCtx’s controller thresholds are method-specific and are selected on validation data before test-time evaluation.                                                                                                           |

Table 11: Comparison protocol used to align shared components while preserving the native behavior of each baseline.