SEGMEM-RAG: Adaptive Memory for Retrieval-Augmented Generation in Open-Ended Knowledge Environments

Xuanbo Fan[1, 2, 3], **Tianqi Zhao**[3], **Yi Cheng**[3], **Chi Xiu**[3], **Jiaxin Guo**[1, 2], **Boci Peng**[1, 2], **Bingjing Xu**[3], **Jessica Zhang**[3], **Feng Sun**[3], **Yan Zhang**[1, 2*]

[1] School of Intelligence Science and Technology, Peking University, Beijing, China
[2] State Key Laboratory of General Artificial Intelligence, Peking University, Beijing, China
[3] Microsoft Corporation, Beijing, China
{xuanbo.fan,guojiaxin,bcpeng}@stu.pku.edu.cn,
{tiazhao,chengyi, chxiu, mirrorxu,jessicaz,sunfeng}@microsoft.com,
zhyzhy001@pku.edu.cn

Abstract

Retrieval-Augmented Generation (RAG) improves the factual accuracy of large language models by grounding responses in external content. However, most RAG systems assume access to static and well-organized corpora with fixed retrieval logic. In practice, real-world sources are heterogeneous and unlabeled, including user-uploaded documents, manuals, and datasets. Effective access in such settings requires adaptive and self-directed retrieval behavior. We present SEGMEM-RAG, a memory-augmented RAG framework that learns to route queries across multiple unlabeled corpora based on experience. It incrementally updates a structured memory and uses self-reflection to guide retrieval over time without supervision. Experimental results demonstrate that SEGMEM-RAG significantly outperforms recent baselines in generation quality on multi-corpus QA tasks.

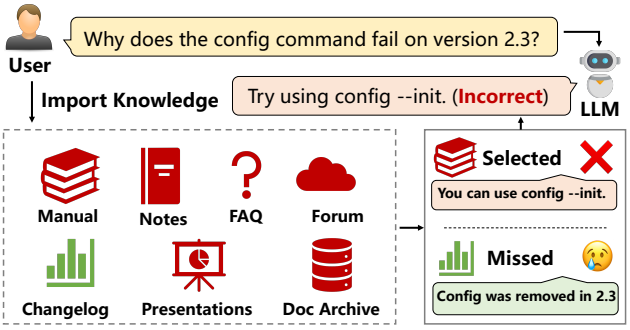

Figure 1: Illustration of incorrect retrieval in open-ended environments: the LLM selects a suboptimal source, resulting in a factually incorrect answer.

Introduction

Retrieval-Augmented Generation (RAG) has emerged as a compelling strategy to improve the factual accuracy of large language models (LLMs) by grounding outputs in external sources(Touvron et al. 2023; Zhao et al. 2023). By retrieving relevant, up-to-date information, RAG reduces hallucinations and enhances response quality across both open-domain and specialized tasks(Arslan et al. 2024).

While RAG systems have seen wide adoption(Fan et al. 2024; Leng et al. 2024), they are typically designed for static and organized corpora with well-defined retrieval configurations. In contrast, real-world environments present a more volatile and loosely structured landscape, where content changes frequently and supervision—such as annotations or metadata—is sparse or outdated. Figure 1 illustrates a typical failure: the system selects an outdated manual while overlooking a more relevant changelog, resulting in a factually incorrect answer.

This scenario poses three fundamental challenges: **First**, users typically interact with a single unified interface, such as a chatbot or enterprise search portal, without specifying which knowledge source to query. The system must therefore autonomously determine where to retrieve from, what to retrieve, and how to organize the results into coherent outputs. **Second**, knowledge sources vary widely in structure, granularity, and style, and new content may be introduced or deprecated over time. Retrieval pipelines with static corpus configurations or hand-crafted source selection rules lack the flexibility to adapt to such heterogeneity. **Third**, as knowledge sources evolve and expand over time, labeled data and query logs quickly become outdated or unavailable. This makes it infeasible to rely on static supervision or hand-crafted retrieval logic.

Prior works has tackled aspects of this problem. Retrieval planners such as ResLLM (Wang et al. 2025) and Omni-RAG (Chen et al. 2025) learn corpus selection over multiple sources, but require retraining when knowledge changes. In-session agents like ReAct (Yao et al. 2023) and Reflexion (Shinn et al. 2023) use step-by-step reasoning, yet lack mechanisms to carry learning across sessions or corpora. Interface-based methods like EasyTool (Yuan et al. 2024) and DRAFT (Qu et al. 2025) rely on tool descriptions or APIs, cannot generalize to open-ended, non-programmatic sources, unfortunately sometimes.

These limitations point to the need for a **self-guided RAG** paradigm, where the system actively plans, monitors, and

*Corresponding author

adjusts its own retrieval and generation process—without relying on fixed rules or external supervision. Specifically, a effective self-guided RAG system demands three key cognitive capabilities: (1) **Localized Planning**: the ability to isolate each retrieval decision from distracting or misleading associations, enabling the system to reason clearly over the relevant decision space. (2) **Retrieval Awareness**: the capacity to reflect on whether the retrieved content genuinely fulfills the intended information need, rather than treating retrieval as a blind, one-shot operation. (3) **Adaptive Learning**: the ability to revise retrieval behavior over time by accumulating feedback from past interactions, gradually promoting reliable sources and suppressing ineffective ones.

To realize these, we introduce SEGMEM-RAG, a modular framework designed for continual adaptation in open-ended knowledge environments. It is carefully designed to support robust, self-guided reasoning through three tightly integrated components:

Segment Planner. Inspired by *Event Segmentation Theory* in cognitive science, this component decomposes the overall reasoning process into localized *segments*, each handling a focused decision step such as selecting a source or composing a subquery. Segment wise reasoning allows the system to operate in bounded contexts, reduce noise from irrelevant memory, and maintain attention on the most salient retrieval decisions.

Feedback Evaluator. Drawing on principles of *generative learning*, this component monitors the outcome of each retrieval segment and generates natural language feedback that reflects success or failure. Rather than relying on external supervision, it performs lightweight self evaluation by contrasting retrieved content with intended information needs, producing signals that inform future planning.

Memory Controller. To support adaptation over time, SEGMEMRAG maintains a symbolic, multi level memory architecture spanning procedural, semantic, and episodic layers. This structured memory encodes retrieval behaviors, source characteristics, and historical outcomes. During inference, it is queried symbolically to guide source selection and strategy refinement, enabling the system to learn from experience without retraining.

Together, these components enable SEGMEM-RAG to plan, reflect, and adapt its own retrieval process—supporting continual self-improvement in dynamic, unlabeled knowledge environments. We validate SEGMEM-RAG on multi-corpus question answering tasks under weak supervision, demonstrating consistent gains over strong retrieval-augmented baselines. Our main contributions are:

- We formalize the problem of retrieval in dynamic, unlabeled knowledge environments where sources are volatile, heterogeneous, and unsupervised sources, and construct a benchmark by composing existing corpora.
- We introduce SEGMEM-RAG, a self-guided RAG framework that integrates segment-wise planning, symbolic feedback, and structured memory.
- We demonstrate that SEGMEM-RAG improves retrieval accuracy and QA quality under weak supervision across diverse multi-corpus benchmarks.

Related Work

Retrieval-Augmented Generation with Multiple Sources

Retrieval-Augmented Generation (RAG) improves the factual grounding of LLMs by retrieving context from external sources (Gao et al. 2023; Fan et al. 2024). However, most RAG systems are designed for static, labeled corpora and struggle in real-world environments where knowledge sources are dynamic, noisy, and unlabeled.

Existing approaches to multi-source RAG typically fall into three categories:

Retrieval planners train corpus selection policies (Wang et al. 2025; Chen et al. 2025), enabling RAG models to select among multiple corpora. While effective in controlled settings, these approaches require retraining when corpora change, limiting scalability in dynamic environments.

In-session agents, such as ReAct and Reflexion (Yao et al. 2023; Shinn et al. 2023), use step-by-step reasoning and tool use within a session. While flexible, they lack mechanisms for cross-session learning or memory, preventing the accumulation of long-term corpus-level knowledge.

Interface-based methods (Yuan et al. 2024; Qu et al. 2025) rely on structured tool or API descriptions to guide retrieval. These systems assume clean, labeled access patterns, which do not generalize to open-ended text corpora without predefined schemas.

In contrast, SEGMEM-RAG treats retrieval as a *self-guided process*: rather than relying on fixed policies, supervision, or structure, it incrementally plans, evaluates, adapts its own retrieval behavior based on feedback—enabling continual adaptation to evolving, unlabeled corpora.

Memory-Augmented Reasoning

A parallel line of work augments LLMs with memory to support long-term reasoning and learning. Some systems perform **in-context reflection**, allowing the model to critique and revise its own outputs (Shinn et al. 2023; Liang et al. 2025; Zhao et al. 2024). Others use **persistent memory** to retain and reuse interaction history across episodes (Xu et al. 2025; Packer et al. 2023; Zhong et al. 2024).

These methods primarily target task-level feedback or dialogue consistency. In contrast, SEGMEM-RAG focuses on memory for *retrieval-level adaptation*—tracking symbolic summaries of retrieval behavior, source utility, and failure patterns over time. Rather than storing past outputs, it maintains a structured memory to shape future planning in volatile, multi-source environments—without supervision or parameter updates.

Preliminaries

We formalize the problem of retrieval-augmented generation in open-ended, multi-source knowledge environments. Let q denote a user query, and let $K = \{K_1, K_2, \ldots, K_N\}$ represent a collection of heterogeneous, unlabeled knowledge sources (e.g., documents, manuals, changelogs). These corpora are diverse in structure, format, and granularity, and lack consistent metadata, interfaces, or annotations.

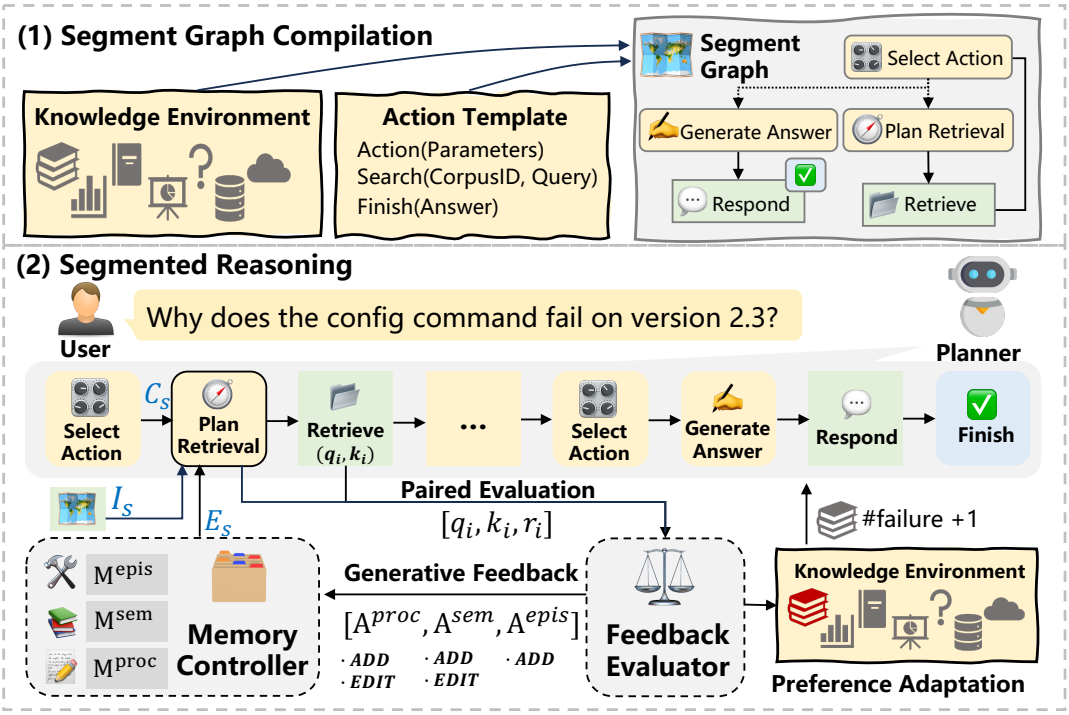

Figure 2: System overview of SEGMEM-RAG. The top shows the compilation of structured action schemas into a segment graph, enabling modular control over retrieval planning. The bottom illustrates segment-wise reasoning in execution: at each step, the agent operates over localized memory, generates symbolic feedback, and updates a structured memory hierarchy. Together, these components support self-guided retrieval and continual adaptation without retraining.

The agent $\mathcal{A}$ must generate an answer to q by planning and executing a sequence of retrieval steps over K. At each step i, the agent selects a source index $k_i \in \{1, \ldots, N\}$, formulates a sub-query q_i, and retrieves content from the selected source:

$$r_i = Retrieve(q_i, K_{k_i}), \tag{1}$$

where r_i denotes the top-ranked passages retrieved. Based on the accumulated retrievals $r_{\leq i}$, the agent generates an intermediate result a_i, which may represent a reasoning step, partial answer, or subgoal resolution.

This interaction produces a trajectory of retrieval and reasoning steps:

$$\tau = \{(q_i, k_i, r_i, a_i)\}_{i=1}^{T}. \tag{2}$$

Unlike traditional systems operating over static or labeled corpora, the agent $\mathcal{A}$ must continually adapt its retrieval planning to maximize answer quality under uncertainty. This includes learning to select relevant sources, avoid redundant retrievals, and structure responses based on evolving query context and prior retrieval outcomes.

Method

Overview

SEGMEM-RAG is a self-guided retrieval-augmented generation framework designed for dynamic, unlabeled knowledge environments. The system integrates planning, reflection, and memory to continually adapt its retrieval behavior—without relying on supervision or static corpus configurations. The framework consists of three tightly coupled components:

Segment Planner. Inspired by event segmentation theory, this component decomposes the reasoning process into modular segments, each focused on a localized decision (e.g., source selection or subquery composition). Segment-wise reasoning allows the system to isolate retrieval steps from irrelevant context, improving precision and enabling symbolic planning over a structured decision graph.

Feedback Evaluator. After each retrieval segment, the system reflects on the outcome by comparing the retrieved content with its intended purpose. It produces natural language feedback and binary success signals, which serve as internal supervision for future planning. This enables the agent to learn from its outcomes without external labels.

Memory Controller. To accumulate experience over time, SEGMEM-RAG maintains a structured symbolic memory with procedural, semantic, and episodic layers. This memory stores interpretable summaries of past retrieval behavior and corpus utility, which are queried during inference to guide source selection and strategy adaptation.

To ensure broad coverage across unfamiliar or underexplored sources, the system includes a lightweight **Cold-start Explorer**, which proactively issues probe queries to bootstrap initial memory signals, facilitating informed planning from the earliest stages of deployment. Together, these components enable SEGMEM-RAG to operate as a

self-monitoring, self-adaptive system capable of navigating open-ended knowledge environments without retraining or supervision. We next detail each component of the system.

Segment Planner

To address the challenges of multi-source retrieval in dynamic environments, SEGMEM-RAG adopts a form of segment-wise reasoning, inspired by *Event Segmentation Theory* (Zacks and Swallow 2007), which suggests that humans naturally partition continuous experience into discrete, meaningful units. The system structures the overall reasoning process into localized decision steps, each executed in an isolated context with access to targeted memory and short-term signals. This modular structure improves focus, reduces contextual interference, and facilitates fine-grained feedback and memory alignment.

Segment Graph Compilation We define a *segment* as a semantically coherent reasoning unit that completes one decision step in a structured task schema. Each segment may operate over one or more arguments, depending on whether those arguments form a tightly coupled reasoning subgoal. For example, selecting both a knowledge source and a retrieval query may be treated as a single segment if they are specified jointly.

To support modular control and symbolic planning, SEGMEM-RAG compiles the agent's structured action schemas into a directed segment graph. This graph represents a finite-state abstraction of the reasoning space, where each node corresponds to a segment state, and edges define valid transitions between segments. The compilation is governed by schema-driven rules that reflect the internal structure and ordering dependencies of available actions.

This construction process is training-free and easily extensible: new tools, corpora, or actions can be introduced by updating the action schema without requiring model retraining or manual rule design. The resulting graph enables localized planning, interpretable trajectories, and context-isolated inference, laying the structural foundation for downstream memory and feedback components.

Segmented Reasoning The agent performs reasoning by traversing the compiled segment graph, making localized decisions at each segment node. During inference, the agent traverses the compiled segment graph by executing one segment at a time. At each segment state s, the system retrieves a micro-instruction I_s, recalls relevant long-term memory entries $E_s = \mathcal{M}.\texttt{recall}(s)$, and extracts localized context $C_s = S.\texttt{relevant}(s)$ from short-term memory S. These inputs are passed to the base model M to compute a decision:

$$o_s = M.\texttt{infer}(I_s,\, E_s,\, C_s) \tag{3}$$

The output o_s is stored in memory and may include structured decisions (e.g., source selection), rationales, or confidence scores. When all required components for an action are available, the agent executes the action (e.g., retrieval), observes the result, and appends it to memory.

Segment transitions are managed by a finite-state machine defined over the action schema, which deterministically advances the planner to the next segment based on filled arguments and observed outcomes. The process continues until an end state is reached or a stopping criterion is met.

This segment-wise execution strategy enables precise control over the reasoning trajectory, limits the working memory scope per step, and supports integration with symbolic memory and feedback evaluators at the segment level.

Algorithm 1: Segment-wise Reasoning in SEGMEM-RAG

Input: Query q, corpora K, agent $\mathcal{A}$ (base model M), memory module $\mathcal{M}$
Output: Final answer $\hat{a}$ with reasoning history

Initialize short-term memory $S \leftarrow \emptyset$
Initialize current state $s \leftarrow \texttt{select\_action}$
Set step counter $t \leftarrow 0$
while $s \neq \texttt{end}$ **and** $t <$ max_steps **do**
  $t \leftarrow t + 1$
  Retrieve micro-instruction I_s for state s
  Recall structured memory $E_s \leftarrow \mathcal{M}.\texttt{recall}(s)$
  Extract working-memory context $C_s \leftarrow S.\texttt{relevant}(s)$
  Infer local decision $o \leftarrow M.\texttt{infer}(I_s, E_s, C_s)$
  Update $S \leftarrow S \cup \{o\}$
  if action parameters are complete **then**
    Execute action (e.g., $\texttt{Retrieve}(q_i, K_{k_i})$) and observe result r
    Append r to S
  end if
  Update state $s \leftarrow \texttt{next}(S)$
end while
return final answer $\hat{a}$ extracted from S

Feedback Evaluator

Generative Feedback After each retrieval segment, SEGMEM-RAG performs a localized self-evaluation to assess whether the retrieved content satisfies the intended purpose of the current sub-query. Instead of modifying model parameters, the system generates symbolic feedback in natural language, which is stored in memory and used to guide subsequent planning decisions.

At step i, the agent issues a sub-query q_i to a selected source K_{k_i}, yielding retrieved content $r_i = Retrieve(q_i, K_{k_i})$. The Feedback Evaluator compares this outcome against the segment's retrieval intent and produces two outputs:

- a binary success label $y_i \in \{0, 1\}$ indicating whether the retrieved passages contain relevant or actionable evidence;
- a descriptive reflection summarizing alignment or mismatch between expectation and observed content.

These signals are asynchronously appended to structured memory $\mathcal{M}$, where they accumulate over time to support long-term utility modeling. Segment-level evaluation ensures timely and localized feedback while avoiding the cost of full-trajectory reflection.

Preference Adaptation To enable retrieval strategies that evolve with experience, SEGMEM-RAG maintains corpus-level failure statistics. For each source K_c, a failure counter f_c is updated whenever a segment involving K_c yields an unsuccessful retrieval ($y_i = 0$). These statistics serve as symbolic priors that bias future source selection—penalizing persistently uninformative corpora while promoting exploration of more reliable alternatives.

This experience-based adaptation mechanism allows the agent to dynamically reweight source preferences without explicit supervision or retraining, improving retrieval robustness in environments with heterogeneous and evolving corpora.

Memory Controller

To support long-term adaptation, SEGMEM-RAG maintains a structured, symbolic memory system $\mathcal{M}$ comprising three cognitively inspired components: *Procedural Memory*, *Semantic Memory*, and *Episodic Memory* (Matthews 2015). Each component stores interpretable natural language summaries derived from prior retrieval episodes and provides symbolic signals to guide future decisions.

Unified Access via Symbolic Similarity. During inference, the Segment Planner queries all memory components in parallel using token-level symbolic similarity between the current sub-query q_i and stored memory entries. This lightweight mechanism supports multilingual usage (e.g., English and Chinese) and avoids reliance on dense embeddings or retraining. Retrieved signals are fused to inform corpus selection and retrieval planning.

Procedural Memory ($\mathcal{M}^{proc}$) captures retrieval heuristics by linking recurring query patterns with preferred sources. For example, "queries about *version conflicts* typically succeed in *technical manuals* (K_1, K_3)." Multiple heuristics may be recalled and aggregated per query. New rules are added when novel behaviors emerge, while existing ones are revised in response to feedback.

Semantic Memory ($\mathcal{M}^{sem}$) maintains corpus-level descriptors summarizing strengths and limitations across query types—such as "strong on configuration troubleshooting," "uses CLI-style expressions," or "frequently omits version identifiers." Each corpus may have multiple descriptors, but only the most similar one is retained per query to estimate source relevance. Reliability statistics (e.g., failure counts) are also maintained to support adaptive planning.

Episodic Memory ($\mathcal{M}^{epis}$) logs symbolic traces of individual retrieval episodes. Each entry records the issued query q_i, selected source K_{k_i}, and a natural language summary of the retrieval outcome. These immutable entries capture concrete precedents—e.g., "query about *X version mismatch* failed on K_2 due to lack of version evidence"—that support case-based reasoning during future segments.

Asynchronous Memory Updates. All memory components are updated in a training-free, heuristic manner based on feedback signals generated at the end of each segment. Updates include rule edits, descriptor refinements, and episodic additions. Importantly, these updates are performed asynchronously after batch inference, ensuring that memory construction does not introduce latency during real-time interaction.

This memory architecture enables SEGMEM-RAG to continually refine its retrieval strategies, model corpus utility under uncertainty, and adapt to emerging usage patterns—all without retraining or reliance on dense representations.

Coldstart Explorer

SEGMEM-RAG incorporates a coldstart mechanism to enable adaptation to under-explored corpora, particularly newly added or long-tail sources lacking sufficient interaction history. This component allows the system to proactively gather initial signals without requiring retraining or supervision.

The routine is triggered when the number of interactions with a corpus K_c falls below a threshold T_{min}. In such cases, the system launches *probe segments*—simulated reasoning steps that sample representative sub-queries from Procedural Memory ($\mathcal{M}^{proc}$) and Episodic Memory ($\mathcal{M}^{epis}$). These probes are issued against K_c, and the outcomes are evaluated to produce binary success signals and descriptive feedback. Resulting signals are incorporated into the symbolic memory $\mathcal{M}$, allowing the system to build experience with underutilized sources and improve planning over time. This process ensures that emerging corpora are actively integrated into the retrieval space, sustaining adaptability in dynamic knowledge environments.

Experimental Setup

Knowledge Environment Setup

We construct a multi-corpus retrieval environment where each knowledge source is independent, unlabeled, and weakly structured. Corpora span diverse domains, languages, and formats. Details are summarized in Table 1.

To retain real-world imbalance, we do not normalize corpus size or coverage. Each corpus is independently indexed using `multilingual-e5-large` (Wang et al. 2024a), chosen for its cross-domain and multilingual robustness.

At inference time, the system may retrieve from any source. For each corpus, the system receives only a numeric index and a brief natural language description automatically generated by GPT-4o (OpenAI 2024). The descriptions are derived from corpus content summaries and are identical across all methods. No downstream task information is used during their construction.

Tasks and Metrics

Unlike standard settings where each QA task is paired with a dedicated corpus, we evaluate in a unified multi-corpus environment shared across all tasks. The system operates over a large pool of heterogeneous knowledge sources without knowing which ones are best suited for each task.

Open-domain QA We evaluate SEGMEM-RAG on multiple datasets, including Natural Questions (Kwiatkowski et al. 2019)(NQ), TriviaQA (Joshi et al. 2017)(Trivia), PopQA (Mallen et al. 2023)(Pop), HotpotQA (Yang et al. 2018)(Hotp), and 2WikiMultiHopQA (Ho et al.

2020)(2wiki). These datasets include both single-hop and multi-hop open-domain QA, requiring retrieval and reasoning over Wikipedia and web documents.

Financial QA We use OmniEval (Wang et al. 2024b), a Chinese financial QA dataset, which consists of four tasks: Extractive QA(Extra), Long-form QA(Long), Multi-hop Reasoning(Multi), and Construct QA(Contr), each targeting different scenarios.

Biomedical QA We use the BioASQ (Tsatsaronis et al. 2015) (Bio) fact-based task, which focuses on retrieving relevant literature from a given question and generating precise entity answers and summary-type answers.

Metrics We use two evaluation metrics: **token-level F1** and **fact-level F1**. Token-level F1 is used on open-domain QA tasks and measures lexical overlap between predicted and reference answers (Jin et al. 2025; Sun et al. 2025). Fact-level F1 is used for financial and biomedical QA, where answers are longer and more varied. Following prior work on LLM-as-a-judge (Chen, Gao, and He 2023), we extract verifiable claims from predictions and references using a large language model, and compute F1 based on their overlap. Prompts are fixed and temperature is set to 0 for consistency.

Baselines

We group baselines by capability: retrieval control, agentic reasoning, memory-based reflection, and oracle settings.

Retrieval Control. Prompting issues fixed natural-language prompts to the retriever without intermediate reasoning or feedback. **BlindRAG** randomly selects a corpus (based only on its ID/description) and retrieves within it.

Agentic Reasoning. ResLLM (Wang et al. 2025) performs description-based corpus routing with a language model. **ReAct** (Yao et al. 2023) interleaves chain-of-thought reasoning with retrieval actions.

Memory and Reflection. Reflexion (Shinn et al. 2023) adds step-wise verbal self-feedback. **DRAFT** (Qu et al. 2025) leverages interaction history to refine guidance and inform subsequent actions.

Oracle. EnumRAG retrieves from *all* corpora for each query (enumeration) and aggregates evidence. **Gold-RAG** queries only the single corpus that achieves the best overall performance (oracle choice), using it for all queries.

Implementation Details

We use `Qwen2.5-7B-Instruct` as the *Reasoning Model*, and `Qwen2.5-32B-Instruct` as the *LLM Judge* for fact-level evaluation. For each query, the system processes up to 32 reasoning segments. Both *Memory Retrieval* and *Knowledge Retrieval* select the top-5 entries from internal memory and external corpora, respectively. We randomly sample 500 QA pairs from each dataset for testing.

Experimental Results

Main Results

Table 2 summarizes the overall performance of all evaluated methods across open-domain, financial, and biomedical QA tasks. We highlight four key observations.

Corpus	Domain/Type	#Docs
arguana (Thakur et al. 2021)	Arg / Snip	8.7k
webis (Thakur et al. 2021)	Arg / Snip	382.5k
nfcorpus (Boteva et al. 2016)	Bio / Docs	3.6k
pubmedqa (Xiong et al. 2024)	Bio / Abs	3.3k
statpearls (Xiong et al. 2024)	Bio / Art	9.6k
textbooks (Xiong et al. 2024)	Bio / Txtbk	125.8k
bioasq (Tsatsaronis et al. 2015)	Bio / Art	40.2k
omnieval (Wang et al. 2024b)	Fin / Rpt	364.8k
dbpedia (Hasibi et al. 2017)	KB / Trip	4.6M
lexrag (Li et al. 2025)	Legal / Law	5.5k
barexam_qa (Zheng et al. 2025)	Legal / Law	1.2k
housing_qa (Zheng et al. 2025)	Legal / Law	1.8M
scidocs (Cohan et al. 2020)	Sci / Docs	25.7k
scifact (Wadden et al. 2020)	Sci / Abs	5.2k
trec-covid (Roberts et al. 2021)	Sci / Art	171.3k
msmarco (Nguyen et al. 2016)	Web / Snip	8.8M
hotpotqa (Yang et al. 2018)	Wiki / Docs	5.2M
nq (Kwiatkowski et al. 2019)	Wiki / Docs	2.7M

Table 1: Corpora used in the multi-source setup. Each corpus is indexed independently. Domain abbreviations: **Arg**umentation, **Bio**medical, **Fin**ancial, **K**nowledge **B**ase, **Legal** Text, **Sci**entific, **Web** Text, **Wiki**pedia. Type abbreviations: Passage **Snip**pets, **Doc**uments, **Abs**tracts, **Art**icles, **Textb**ooks, **Report**s, Knowledge **Trip**les.

SEGMEM-RAG tends to outperform strong baselines and even oracle methods. Our method achieves the highest average performance across all three domains, outperforming strong baselines such as ReAct, Reflexion, and even oracle-style methods EnumRAG and Gold-RAG. In particular, SEGMEM-RAG excels on challenging multi-hop datasets (HotpotQA and 2WikiMultiHopQA), showing clear advantages in retrieving relevant evidence. Though certain individual tasks see competitive results from oracle methods, SEGMEM-RAG consistently provides the best overall balance across diverse tasks and domains.

Accurate corpus selection is crucial in multi-source environments. EnumRAG significantly outperforms Prompting, confirming that effective retrieval enhances performance. In contrast, BlindRAG, which randomly selects sources, consistently underperforms across tasks, demonstrating that naive corpus selection introduces harmful noise and disrupts reasoning. Robust corpus selection thus emerges as essential but challenging in heterogeneous knowledge environments.

Simple description-based retrieval is insufficient. Methods relying solely on static corpus descriptions (e.g., ResLLM and BlindRAG) consistently perform poorly, with results substantially below Prompting. Even ReAct, which adds intermediate reasoning steps, achieves limited improvement. This suggests that static summaries and shallow interactions are inadequate for effective retrieval in complex multi-source environments.

Type	Method	Open-Domain						Financial					Medical
		NQ	Trivia	Pop	Hotp	2wiki	Avg.	Extra	Long	Multi	Constr	Avg.	Bio
Retrieval Control	Prompting	25.6	44.2	31.4	23.0	26.3	30.1	12.4	12.6	14.5	9.2	12.2	21.0
	BlindRAG	12.1	30.0	14.5	10.3	11.7	15.7	5.9	19.2	15.1	18.7	14.7	12.4
Agentic Reasoning	ResLLM	19.2	29.0	3.2	11.5	9.3	14.4	21.2	26.0	27.3	25.5	25.0	15.5
	React	15.2	34.4	14.3	11.2	9.9	17.0	19.4	20.9	22.7	22.6	21.4	36.4
Memory & Reflection	Reflection	26.5	39.1	12.1	18.8	20.2	23.4	14.1	17.8	23.9	22.4	19.5	42.0
	DRAFT	42.2	62.4	35.3	39.3	31.2	42.1	**36.4**	34.1	28.5	30.9	32.5	39.5
Oracle	EnumRAG	47.1	61.3	**64.2**	32.7	23.7	45.8	33.3	31.1	30.3	29.9	31.1	31.4
	GoldRAG	**50.6**	53.8	58.2	32.8	16.7	42.4	33.9	29.7	30.9	31.2	31.2	38.2
Self-Guide	SEGMEM-RAG	47.8	**64.2**	54.2	**40.0**	**31.4**	**47.5**	35.9	**35.1**	**33.9**	**31.4**	**34.1**	**44.0**

Table 2: Overall comparison of SEGMEM-RAG with baseline methods across three task domains. Token-level F1 is reported for Open-domain QA, while Fact-level F1 (LLM-judged) is reported for Financial QA and Biomedical QA (Medical). Best-performing methods per task are highlighted in bold.

Feedback-driven retrieval surpasses exhaustive and gold-corpus strategies. Surprisingly, SEGMEM-RAG, leveraging segment-wise feedback and structured memory, outperforms oracle-style strategies such as exhaustive enumeration (EnumRAG) and optimal single-corpus retrieval (Gold-RAG). This demonstrates the advantage of targeted feedback mechanisms over brute-force enumeration and fixed corpus selection constraints.

Ablation Study

To assess the contribution of each core module in SEGMEM-RAG, we conduct an ablation study by selectively disabling individual components while holding all other configurations constant. Table 3 reports token-level F1 scores on two representative datasets: NQ and PopQA.

Segment Planner provides essential structure for multi-step retrieval. Removing the Segment Planner leads to consistent performance degradation on both tasks, with a particularly large drop on PopQA (–30.6%). This confirms the importance of segment-wise control in decomposing reasoning into manageable sub-decisions and maintaining contextual focus across steps. Without segmentation, the agent is more prone to semantic drift and redundant retrievals.

Feedback Evaluator supports self-correction and signal alignment. Disabling the Feedback Evaluator reduces performance by 9.6% on NQ and 18.2% on PopQA. These results highlight its role in assessing retrieval quality at each step and providing actionable feedback to guide subsequent planning. Without reflection, the system struggles to realign when intermediate retrievals diverge from query intent.

Memory Controller enables long-range adaptation. The largest performance loss occurs when the Memory Controller is removed, with F1 scores dropping by 33.7% (NQ) and 40.4% (PopQA). This underscores the value of symbolic memory in capturing corpus-level utility patterns and informing future decisions. In its absence, the system reverts to short-sighted selection, failing to learn from prior retrieval outcomes.

Method	NQ	%↓	PopQA	%↓
SEGMEM-RAG	**47.8**	–	**54.2**	–
w/o Segment Planner	45.4	5.0%	37.6	30.6%
w/o Feedback Evaluator	43.2	9.6%	44.4	18.2%
w/o Memory Controller	31.7	**33.7%**	32.3	**40.4%**
w/o Coldstart Explorer	43.3	9.3%	49.0	9.5%

Table 3: Ablation study on SEGMEM-RAG based on NQ and PopQA. Scores reported as token-level F1. Percentages indicate performance drops compared to full system.

Coldstart Explorer improves coverage of low-usage sources. Eliminating the Coldstart Explorer results in a moderate drop on both datasets (–9.3% and –9.5%, respectively), indicating its contribution to bootstrapping memory for underexplored corpora. While less critical in balanced benchmark settings, this module is expected to yield greater gains in more dynamic environments, such as with newly added or long-tail sources.

Conclusion

In this paper, we propose SEGMEM-RAG, a self-guided approach to retrieval-augmented generation in dynamic, unlabeled knowledge environments. The system adapts its retrieval behavior based on internal evaluation and prior interactions, without relying on supervision or retraining. Our approach supports three key capabilities for open-ended retrieval: localized planning to reduce context interference, retrieval awareness to assess evidence quality, and adaptive learning to improve decisions over time.

Empirical results across multi-source benchmarks suggest that self-guided retrieval can enhance factual accuracy and contextual relevance. These findings highlight its potential as a foundation for more resilient and adaptive language systems in evolving knowledge settings.

Acknowledgments

This work is supported in part by Ucap Cloud.

References

Arslan, M.; Ghanem, H.; Munawar, S.; and Cruz, C. 2024. A Survey on RAG with LLMs. *Procedia computer science*, 246: 3781–3790.

Boteva, V.; Gholipour, D.; Sokolov, A.; and Riezler, S. 2016. A full-text learning to rank dataset for medical information retrieval. In *European Conference on Information Retrieval*, 716–722. Springer.

Chen, S.; Gao, S.; and He, J. 2023. Evaluating Factual Consistency of Summaries with Large Language Models. arXiv:2305.14069.

Chen, Z.; Liao, Y.; Jiang, S.; Wang, P.; Guo, Y.; Wang, Y.; and Wang, Y. 2025. Towards Omni-RAG: Comprehensive Retrieval-Augmented Generation for Large Language Models in Medical Applications. *arXiv preprint arXiv:2501.02460*.

Cohan, A.; Feldman, S.; Beltagy, I.; Downey, D.; and Weld, D. S. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *ACL*.

Fan, W.; Ding, Y.; Ning, L.; Wang, S.; Li, H.; Yin, D.; Chua, T.-S.; and Li, Q. 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*, 6491–6501.

Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, H.; and Wang, H. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

Hasibi, F.; Nikolaev, F.; Xiong, C.; Balog, K.; Bratsberg, S. E.; Kotov, A.; and Callan, J. 2017. DBpedia-Entity V2: A Test Collection for Entity Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, 1265–1268. ACM.

Ho, X.; Duong Nguyen, A.-K.; Sugawara, S.; and Aizawa, A. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In Scott, D.; Bel, N.; and Zong, C., eds., *Proceedings of the 28th International Conference on Computational Linguistics*, 6609–6625. Barcelona, Spain (Online): International Committee on Computational Linguistics.

Jin, B.; Zeng, H.; Yue, Z.; Yoon, J.; Arik, S.; Wang, D.; Zamani, H.; and Han, J. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Joshi, M.; Choi, E.; Weld, D.; and Zettlemoyer, L. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In Barzilay, R.; and Kan, M.-Y., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1601–1611. Vancouver, Canada: Association for Computational Linguistics.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Kelcey, M.; Devlin, J.; Lee, K.; Toutanova, K. N.; Jones, L.; Chang, M.-W.; Dai, A.; Uszkoreit, J.; Le, Q.; and Petrov, S. 2019. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics*.

Leng, Q.; Portes, J.; Havens, S.; Zaharia, M.; and Carbin, M. 2024. Long context rag performance of large language models. *arXiv preprint arXiv:2411.03538*.

Li, H.; Chen, Y.; Hu, Y.; Ai, Q.; Chen, J.; Yang, X.; Yang, J.; Wu, Y.; Liu, Z.; and Liu, Y. 2025. LexRAG: Benchmarking Retrieval-Augmented Generation in Multi-Turn Legal Consultation Conversation. arXiv:2502.20640.

Liang, X.; Tao, M.; Xia, Y.; Wang, J.; Li, K.; Wang, Y.; He, Y.; Yang, J.; Shi, T.; Wang, Y.; et al. 2025. SAGE: Self-evolving Agents with Reflective and Memory-augmented Abilities. *Neurocomputing*, 130470.

Mallen, A.; Asai, A.; Zhong, V.; Das, R.; Khashabi, D.; and Hajishirzi, H. 2023. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 9802–9822. Toronto, Canada: Association for Computational Linguistics.

Matthews, B. R. 2015. Memory dysfunction. *CONTINUUM: Lifelong Learning in Neurology*, 21(3): 613–626.

Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; and Deng, L. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *CoRR*, abs/1611.09268.

OpenAI. 2024. GPT-4o System Card. arXiv:2410.21276.

Packer, C.; Fang, V.; Patil, S.; Lin, K.; Wooders, S.; and Gonzalez, J. 2023. MemGPT: Towards LLMs as Operating Systems.

Qu, C.; Dai, S.; Wei, X.; Cai, H.; Wang, S.; Yin, D.; Xu, J.; and Wen, J.-R. 2025. From Exploration to Mastery: Enabling LLMs to Master Tools via Self-Driven Interactions. In *The Thirteenth International Conference on Learning Representations*.

Roberts, K.; Alam, T.; Bedrick, S.; Demner-Fushman, D.; Lo, K.; Soboroff, I.; Voorhees, E.; Wang, L. L.; and Hersh, W. R. 2021. Searching for Scientific Evidence in a Pandemic: An Overview of TREC-COVID. arXiv:2104.09632.

Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K. R.; and Yao, S. 2023. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Sun, H.; Qiao, Z.; Guo, J.; Fan, X.; Hou, Y.; Jiang, Y.; Xie, P.; Huang, F.; and Zhang, Y. 2025. ZeroSearch: Incentivize the Search Capability of LLMs without Searching. arXiv:2505.04588.

Thakur, N.; Reimers, N.; Rücklé, A.; Srivastava, A.; and Gurevych, I. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.;

Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Tsatsaronis, G.; Balikas, G.; Malakasiotis, P.; Partalas, I.; Zschunke, M.; Alvers, M. R.; Weissenborn, D.; Krithara, A.; Petridis, S.; Polychronopoulos, D.; et al. 2015. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16: 1–28.

Wadden, D.; Lin, S.; Lo, K.; Wang, L. L.; van Zuylen, M.; Cohan, A.; and Hajishirzi, H. 2020. Fact or Fiction: Verifying Scientific Claims. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 7534–7550. Online: Association for Computational Linguistics.

Wang, L.; Yang, N.; Huang, X.; Yang, L.; Majumder, R.; and Wei, F. 2024a. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Wang, S.; Tan, J.; Dou, Z.; and Wen, J.-R. 2024b. OmniEval: An Omnidirectional and Automatic RAG Evaluation Benchmark in Financial Domain. arXiv:2412.13018.

Wang, S.; Zhuang, S.; Koopman, B.; and Zuccon, G. 2025. Resllm: Large language models are strong resource selectors for federated search. In *Companion Proceedings of the ACM on Web Conference 2025*, 1360–1364.

Xiong, G.; Jin, Q.; Lu, Z.; and Zhang, A. 2024. Benchmarking Retrieval-Augmented Generation for Medicine. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics: ACL 2024*, 6233–6251. Bangkok, Thailand: Association for Computational Linguistics.

Xu, W.; Liang, Z.; Mei, K.; Gao, H.; Tan, J.; and Zhang, Y. 2025. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K. R.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations*.

Yuan, S.; Song, K.; Chen, J.; Tan, X.; Shen, Y.; Kan, R.; Li, D.; and Yang, D. 2024. Easytool: Enhancing llm-based agents with concise tool instruction. *arXiv preprint arXiv:2401.06201*.

Zacks, J. M.; and Swallow, K. M. 2007. Event segmentation. *Current directions in psychological science*, 16(2): 80–84.

Zhao, A.; Huang, D.; Xu, Q.; Lin, M.; Liu, Y.-J.; and Huang, G. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19632–19642.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Zheng, L.; Guha, N.; Arifov, J.; Zhang, S.; Skreta, M.; Manning, C. D.; Henderson, P.; and Ho, D. E. 2025. A Reasoning-Focused Legal Retrieval Benchmark. In *Proceedings of the Symposium on Computer Science and Law on ZZZ*, CSLAW '25, 169–193. ACM.

Zhong, W.; Guo, L.; Gao, Q.; Ye, H.; and Wang, Y. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19724–19731.