BubbleRAG: Interactive Cognitive Offloading with Thought Bubble in Retrieval-Augmented Generation

Fuda Ye[1], Jiachuan Wang[2,*], Yongqi Zhang[1], Lei Chen[1,3], Shuangyin Li[4,*]

[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]University of Tsukuba
[3]The Hong Kong University of Science and Technology
[4]South China Normal University

fye374@connect.hkust-gz.edu.cn, wangjc@slis.tsukuba.ac.jp, shuangyinli@scnu.edu.cn

Abstract

Retrieval-augmented generation (RAG) extends the capabilities of large language models (LLMs) by providing access to external knowledge. However, traditional retrieval-augmented LLMs rely on a silent reading paradigm that processes all retrieved documents passively, forcing them to reason without any interaction with the documents. This paradigm contrasts sharply with human interactive reading behavior, where external tools, such as bookmarks and notes, are used to offload cognitive demands. This paper introduces BubbleRAG, an enhanced RAG framework that emulates human interactive reading through annotation and re-reading. Specifically, BubbleRAG utilizes a lightweight thought bubble module that offloads LLM's internal cognition into external bookmark tokens, which are then annotated back into the context. These bookmarks serve as externalized memory, allowing the LLM to revisit these annotations in subsequent reading and answering. Notably, BubbleRAG is particularly suitable for low-resource scenarios, as the LLM parameters remain frozen. Extensive experiments confirm the effectiveness, robustness, and generalizability of BubbleRAG. Our findings demonstrate that BubbleRAG enables LLMs to achieve superior evidence identification abilities typically seen in retrievers, while establishing a cognitive link between external and internal information during answer generation. The source code is available at https://github.com/yefd/BubbleRAG.

1 Introduction

Retrieval-augmented generation (RAG) enhances large language models (LLMs) by integrating retrieval components, granting them access to relevant external knowledge (Lewis et al., 2020; Guu et al., 2020). By leveraging non-parametric, up-to-date information, RAG allows LLMs to extend their knowledge boundaries without additional training. The coordination between retrieval and generation facilitates more accurate responses, particularly in knowledge-intensive domains (Cai et al., 2024).

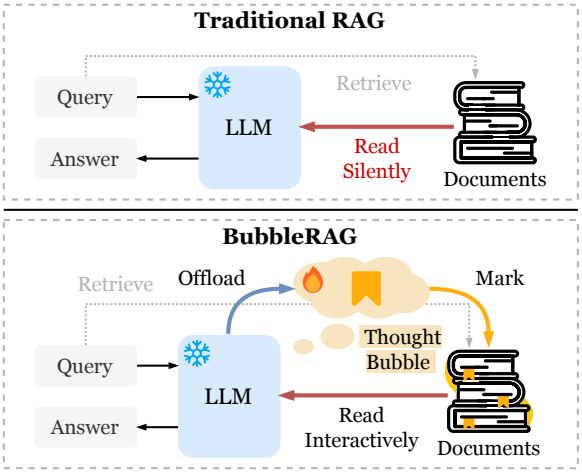

Figure 1: Comparison between traditional RAG and BubbleRAG. ***Top***: RAG processes retrieved documents in a silent reading paradigm. ***Bottom***: BubbleRAG enables interactive reading inspired by human cognitive behavior, allowing the LLM to offload its internal cognition into external bookmarks via a thought bubble.

Specifically, RAG requires LLMs to answer questions based on retrieved documents. In prevailing implementations, all documents are appended to the instruction and processed in a single forward pass, without any interaction with the documents themselves (Gao et al., 2023). While recent efforts primarily focus on enhancing retrieval accuracy or generation capability (Zhu et al., 2025b), these methods continue to inherit the silent reading paradigm that constructs understanding internally.

However, such a silent reading paradigm of current RAG constrains the full potential of LLMs. **First**, LLMs face difficulty in effectively exploiting additional knowledge. LLMs must filter noisy results and identify relevant passages in a global view. However, positional bias and attention diffusion

*Corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2026, pages 43540–43558

cause relevant evidence buried mid-context to be eclipsed by irrelevant details. Consequently, LLMs struggle to detect nuanced relationships among documents and are susceptible to the "lost-in-the-middle" issue (Zhu et al., 2024; Liu et al., 2023). **Secondly**, LLMs are burdened with the memory of reading processes. Retrieval-augmented LLMs are required to process all retrieved documents simultaneously, while retaining all intermediate information in their mind. Even when the correct evidence is identified internally, this memory bottleneck hampers LLMs' abilities to exploit that information during generation (Liu et al., 2025).

Moreover, the silent reading paradigm is contrary to the cognitive processes observed in human reading behavior. In detail, cognitive psychology theories show that humans rarely process texts passively; instead, they frequently employ external tools, such as highlights and notes, to offload cognitive demands onto external representations (Johnson and Shaw, 2008; Liu and Stasko, 2010; Kirsh, 2010). These epistemic actions reduce cognitive load, guide re-reading, and help structure and synthesize information. Therefore, this interactive reading paradigm facilitates deeper comprehension compared to relying solely on internal cognition (Clark, 2010; Risko and Gilbert, 2016; Sweller et al., 2019). In contrast, current RAG reads in silence and lacks such functionality, potentially impairing the performance. Thus, integrating an interactive reading mechanism into RAG that mimics cognitive offloading offers a promising solution.

Following the above idea, in this paper, we propose BubbleRAG, an enhanced RAG framework that employs an interactive reading mechanism inspired by human bookmarking practices. Specifically, BubbleRAG integrates a lightweight thought bubble module that projects the LLM's internal cognition from its hidden state into external bookmark tokens. These tokens are then embedded into the retrieved documents and serve as externalized memory, enabling the LLM to revisit its own previously identified insights. This interactive cognitive offloading reduces the cognitive burden of the LLM, facilitating more precise comprehension of the provided evidence. Overall, our contributions are threefold:

- We introduce BubbleRAG, the first enhanced RAG framework that breaks the silent reading paradigm by enabling LLMs with the interactive reading capability.
- A lightweight thought bubble module is designed to convert the LLM's internal cognitive states into external bookmark tokens, seamlessly injecting reasoning traces back into the retrieved context without LLM re-training.
- Extensive experiments demonstrate that BubbleRAG consistently outperforms baseline methods in terms of effectiveness, robustness, and generalizability, highlighting its potential to enhance the capabilities of retrieval-augmented LLMs significantly.

2 Related Work

2.1 Retrieval-Enhanced RAG

The effectiveness of RAG pipelines is fundamentally constrained by the quality of documents returned by the retriever (Ren et al., 2025; Xie et al., 2023). Consequently, enhancing retrieval accuracy is a central focus of recent research. Early efforts target improvements to the retriever itself, such as jointly tuning to maximize the end-to-end performance (Shi et al., 2023). However, joint optimization can be computationally prohibitive, especially when dealing with LLMs comprising billions of parameters. More efficient alternatives are to keep the LLM frozen and update only the retriever or incorporate a separate re-ranker (Shi et al., 2024; Bai et al., 2023). Recent efforts also explore adaptive (Jeong et al., 2024; Li et al., 2026a), iterative (Shao et al., 2023; Baek et al., 2025), and recursive (Kim et al., 2023; Wang et al., 2024b) retrieval strategies that balance knowledge coverage and retrieval latency. In addition, document-level truncation and selection techniques have proven to be straightforward yet effective methods for refining candidate lists (Gao et al., 2023; Xu et al., 2024; Yan et al., 2024). Other approaches utilize small language models to refine the contents in retrieved documents, reducing noise and computational complexity (Jiang et al., 2024; Pan et al., 2024).

2.2 Generation-Enhanced RAG

Where retrieval-enhanced approaches focus on improving the quality of evidence, generation-enhanced methods equip the LLMs to utilize the retrieved evidence (Zhao et al., 2024). Prompt engineering is a widely adopted technique to augment the LLMs without training, which is suitable for RAG scenarios (Lazaridou et al., 2022; Ram et al., 2023; Sahoo et al., 2024). Further improvements can be achieved through fine-tuning LLMs. For example, Yoran et al. (2024) utilize

instruction tuning to equip the LLM with counterfactual resistance ability. Some joint modeling approaches (Sachan et al., 2021; Glass et al., 2022; Izacard et al., 2023) optimize both the retriever and the LLM, but they compromise the generalization capabilities of LLMs (Gekhman et al., 2024; Han et al., 2025). Alternative approaches focus on augmenting the LLM's abilities by introducing additional trainable components. For example, Zhu et al. (2025a) train a set of learnable virtual tokens to enhance the LLM's performance. More recently, reasoning-augmented RAG methods have emerged to explicitly guide the generation process through structured reasoning or self-reflection (Zhao and Li, 2025; Li et al., 2026b). LLM-driven approaches, such as Self-RAG (Asai et al., 2024) and Self-Reasoning RAG (Xia et al., 2024), enable models to iteratively reflect on intermediate reasoning states and decide whether retrieval is required. Graph-driven methods, including HippoRAG (Gutiérrez et al., 2024) and TRACE (Fang et al., 2024), construct structured knowledge representations from documents to facilitate reasoning. These approaches improve reasoning robustness by introducing explicit control. However, the feedback signals in these methods are typically generated alongside reasoning and remain external to the retrieved documents themselves.

Despite these advances, most existing RAG methods still follow a silent reading paradigm, forcing LLMs to process and memorize a large amount of information and limiting their capacity to fully exploit relevant evidence. In this work, we propose BubbleRAG that introduces interactive reading to externalize intermediate cognition in a practical and non-intrusive way.

3 BubbleRAG

3.1 Problem Formulation and Overview

RAG enhances LLMs by incorporating external documents into generation. For simplicity, we abstract away tokenization and embedding and assume all inputs are represented in the input embedding space. Formally, an LLM generator $\mathcal{G}(\cdot)$ receives a prompt $\mathbf{P}$ that combines a query $\mathbf{q}$ with a set of top-k retrieved documents $\mathbf{D} = \{\mathbf{d}_i\}_{i=1}^{k}$. The prompt construction process is defined as:

$$\mathbf{P} = \text{Concat}(\mathbf{q}, \mathbf{D}), \tag{1}$$

where Concat denotes concatenation within a predefined prompt template. Then, we define a reading function that maps the prompt to hidden states:

$$f_{\text{read}}(\mathbf{P}) \triangleq \mathcal{G}_{\text{att}}(\mathbf{P}), \tag{2}$$

where $\mathcal{G}_{\text{att}}$ is the attention layers. The resulting hidden states are $\mathbf{H} \in \mathbb{R}^{|\mathbf{P}| \times h_{\mathcal{G}}}$, where $h_{\mathcal{G}}$ is the dimension of LLM's input space. We next define the answering function, which performs autoregressive decoding based on the hidden states:

$$f_{\text{answer}}(\mathbf{H}) \triangleq \arg\max_{\boldsymbol{a}} \prod_{t=1}^{|\boldsymbol{a}|} p_{\mathcal{G}}(a_t \mid \mathbf{H}, \boldsymbol{a}_{<t}), \tag{3}$$

where $p_{\mathcal{G}}$ is the next-token distribution modeled by the LLM, and $\boldsymbol{a}_{<t}$ represents the answer tokens preceding position t.

In traditional RAG frameworks, LLMs operate under a silent reading paradigm, in which all retrieved documents are processed internally before producing a final answer $\hat{\boldsymbol{a}}$. This traditional RAG pipeline is expressed as:

$$\hat{\boldsymbol{a}} = f_{\text{answer}}(\mathbf{H}), \mathbf{H} = f_{\text{read}}(\mathbf{P}). \tag{4}$$

In contrast, Figure 2 illustrates our BubbleRAG framework. Initially, BubbleRAG aims to generate bookmarks that are inserted back into each document through an interactive reading process. The objective is to obtain a processed set of documents with bookmarks, denoted as $\mathbf{D}^{\star} = \{\mathbf{d}_i^{\star}\}_{i=1}^{k}$. We reformulate this interactive reading as an autoregressive process, analogous to the step-by-step reading behavior of humans. At each step, the internal cognition of LLM will be projected into external representations – bookmark tokens.

In the inference stage, the interactive reading process can be accelerated using key-value (KV) caching, enabling a favorable balance between efficiency and effectiveness. Importantly, the backbone LLM is kept frozen, making BubbleRAG particularly well-suited for low-resource scenarios.

3.2 Thought Bubble

Recall that LLMs struggle to offload their cognition during the silent reading process. Inspired by previous work on latent reasoning (Hao et al., 2025), and instead of fine-tuning the LLM itself, we propose the thought bubble as a trainable module to convert the LLM's internal cognition into external representations. As shown in the right part of Figure 2, the thought bubble is a lightweight, pluggable encoder that operates outside the LLM. Thought bubble accepts the intermediate hidden states of LLM as inputs and outputs external bookmark tokens.

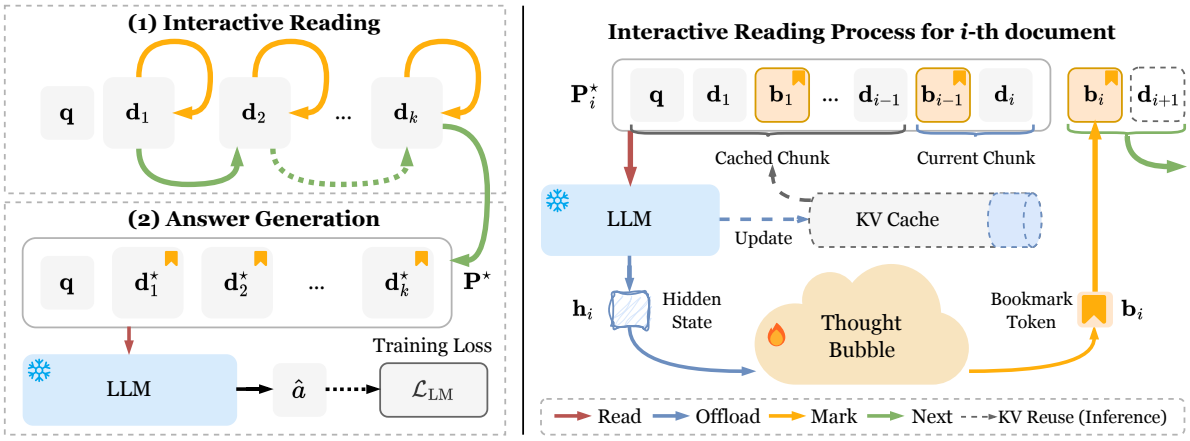

Figure 2: Overview of the BubbleRAG. ***Left***: The pipeline includes: (1) Interactive reading, where the LLM reads documents and appends bookmarks; and (2) Answer generation, where the LLM uses the marked documents to generate an answer. ***Right***: The interactive reading process for the i-th document, where hidden states are converted into bookmark tokens by the thought bubble. During inference, KV caching enables reuse of computed states.

Specifically, we utilize the hidden state of the last token from the LLM's last attention layer to represent its internal cognition at each reading step, which contains rich intermediate information (Zhang et al., 2025). When the LLM reads i-th document $\mathbf{d}_i$, together with the query $\mathbf{q}$ and previous marked documents $\mathbf{D}^\star_{<i}$, the input prompt for generating the i-th bookmark is constructed by:

$$\mathbf{P}^\star_i = \text{Concat}\left(\mathbf{q}, \mathbf{D}^\star_{<i}, \mathbf{d}_i\right). \tag{5}$$

As a standard structure, the LLM processes the inputs to yield the intermediate state:

$$\mathbf{h}_i = f_{\text{read}}(\mathbf{P}^\star_i)[\text{-}1], \tag{6}$$

where $\mathbf{h}_i \in \mathbb{R}^{1\times h_\mathcal{G}}$ is the last token's hidden state.

The thought bubble, a two-layer fully-connected encoder denoted as $f_{\text{bubble}}(\cdot)$, projects this hidden state into the LLM's input embedding space, an external yet accessible space for the LLM. Specifically, the thought bubble produces a continuous bookmark token $\mathbf{b}_i \in \mathbb{R}^{1\times h_\mathcal{G}}$, which is defined as:

$$\mathbf{b}_i = f_{\text{bubble}}\left(\mathbf{h}_i\right). \tag{7}$$

Then, this bookmark token is embedded to the end of the document, serving as an external note for subsequent reading and answer generation. The mark process is defined by:

$$\mathbf{d}^\star_i = \mathbf{d}_i \oplus \mathbf{b}_i, \tag{8}$$

where $\oplus$ is concatenation in the embedding space.

Notably, bookmark tokens do not correspond to any actual vocabulary tokens, but instead serve as externalized memory that the LLM can utilize. By leveraging these bookmark tokens, the LLM moves beyond the silent reading paradigm: the LLM can offload its cognition on each document, leave behind annotations for itself, and integrate these insights during future answer generation.

After completing the interactive reading process, all documents $\mathbf{D}^\star$ are annotated with their respective bookmarks. The LLM then generates the final response using these annotated documents:

$$\hat{\boldsymbol{a}} = f_{\text{answer}}\left(\mathbf{H}^\star\right), \mathbf{H}^\star = f_{\text{read}}\left(\mathbf{P}^\star\right), \tag{9}$$

where $\mathbf{P}^\star$ denotes the final input embedding.

For clarity, the above formulation simplifies the input prompt. Comprehensive descriptions and the prompt templates used in our implementation are provided in Appendix D.3.

3.3 Training and Inference Strategy

We adopt a language modeling task, training the LLM to generate subsequent tokens based on the preceding context. The language modeling loss $\mathcal{L}_{\text{LM}}$ is defined as the negative log-likelihood of the target answer sequence. The loss is given by:

$$\mathcal{L}_{\text{LM}} = -\sum_{t=1}^{|\boldsymbol{a}|} \log p_\mathcal{G}(\boldsymbol{a}_t|\mathbf{P}^\star, \boldsymbol{a}_{<t}). \tag{10}$$

During inference, the interactive reading process in BubbleRAG is implemented as an autoregressive procedure, similar to standard token-by-token generation in LLMs. This sequential nature enables the use of KV caching, which allows previously computed hidden states corresponding to cached document chunks to be efficiently reused at each

step, significantly improving inference efficiency. Notably, the backbone LLM is kept frozen, making BubbleRAG highly flexible and particularly well-suited for low-resource scenarios.

The pseudocode is presented in Algorithm 1.

4 Experiments

4.1 Datasets and Metrics

We conduct experiments on four datasets to assess the effectiveness of BubbleRAG. The evaluation covers **Natural Questions (NQ)** (Kwiatkowski et al., 2019), which consists of questions with human-annotated answers sourced from Wikipedia, as well as three multi-hop reasoning datasets based on Wikipedia: **HotpotQA** (Yang et al., 2018), **2WikiMultiHopQA (2Wiki)** (Ho et al., 2020), and **MuSiQue** (Trivedi et al., 2021). For NQ, we adopt the NQ-10 version standardized by Liu et al. (2023), which contains 10 candidate documents per question. HotpotQA and 2Wiki are accompanied by 10 candidate documents, while MuSiQue provides 20 documents. Detailed dataset descriptions and statistics are provided in Appendix D.1.

Following previous research (Mallen et al., 2023; Liu et al., 2023; Ye et al., 2024), we adopt accuracy (Acc.) and token-level F1 score as metrics.

4.2 Baselines and Implementation Details

To comprehensively evaluate BubbleRAG, we compared its performance against both standard and enhanced RAG methods across various LLM backbones. For the standard RAG baseline, we select a widely used open-source LLM: Mistral$_{7B}$ ("Mistral-7B-Instruct-v0.3") (Jiang et al., 2023). In addition, we include GPT-4o (Achiam et al., 2023), DeepSeek-V3 (DeepSeek-AI et al., 2024), and DeepSeek-R1 (DeepSeek-AI et al., 2025) as baselines. For enhanced RAG baselines, we categorize them into two groups: **(1) Retrieval-enhanced RAG**, including Adaptive RAG (Jeong et al., 2024), LongLLMLingua (Jiang et al., 2024), LLMLingua-2 (Pan et al., 2024) and BGM (Ke et al., 2024); and **(2) Generation-enhanced RAG**, including ThoT (Zhou et al., 2023), SelfElicit (Liu et al., 2025), MetaRAG (Zhou et al., 2024), Prompt Tuning (Lester et al., 2021), Prefix Tuning (Li and Liang, 2021), SPRING (Zhu et al., 2025a), and R^{2}AG (Ye et al., 2024). All enhanced methods are implemented on the same base LLM for fair comparison. Detailed descriptions and implementations are provided in Appendix D.2.

Although the distractors in four datasets are carefully selected and ranked, we further employ Contriever (Izacard et al., 2021) as the re-ranker to achieve optimal retrieval performance, simulating real-world scenarios. Cosine similarity is used as the scoring function for ranking. Retrieval performance for each dataset is summarized in Table 8.

For BubbleRAG, we set the learning rate to 2×10^{-4}, the maximum gradient norm to 0.3, and the warmup ratio to 0.03. Thought bubble is implemented as a two-layer MLP with a hidden size of $2 \times h_{\mathcal{G}}$. Optimization is performed using Adam (Kingma and Ba, 2014). The training process is completed in a single epoch, ensuring both efficiency and low computational cost.

4.3 Main Results

Table 1 presents the main results, from which we can draw the following conclusions:

(1) BubbleRAG brings substantial improvements for Mistral$_{7B}$. On three multi-hop QA datasets (HotpotQA, 2Wiki, and MuSiQue), BubbleRAG significantly improves both accuracy and F1. These results indicate that BubbleRAG helps LLMs more precisely exploit retrieved documents using interactive reading, yielding more accurate answers.

(2) Compared with closed-source LLMs using standard RAG, BubbleRAG shows competitive ability. With Mistral$_{7B}$ as its backbone, BubbleRAG narrows the accuracy gap to GPT-4o and delivers a higher F1. However, larger LLMs such as DeepSeek-R1 remain strong in accuracy, especially on the longer-context MuSiQue dataset.

(3) It is clear that BubbleRAG consistently outperforms other enhanced RAG methods. (a) Compared with retrieval-enhanced RAG methods, BubbleRAG significantly outperforms them. While methods like Adaptive RAG sometimes gain increases in accuracy metric, all of them suffer a significant drop in F1 score. (b) In training-free reasoning methods, techniques such as ThoT offer modest gains with Mistral$_{7B}$. Our extended experiments on other LLMs in Appendix C.1 highlight that reasoning methods may not significantly improve performance for weaker LLMs (Wei et al., 2022), and LLMs struggle to judge the correctness of their reasoning using their inherent knowledge alone (Huang et al., 2024). (c) Compared with PEFT approaches (SPRING, Prompt Tuning, Prefix Tuning, and R^{2}AG), BubbleRAG still shows competitive performance across most metrics.

	Dataset(→)	NQ-10		HotpotQA		2Wiki		MuSiQue		Avg. Ranking	
	Method(↓)/Metric(→)	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
	GPT-4o†	0.6949	0.4496	0.7503	0.6615	0.8010	0.6465	0.6174	0.4430	-	-
	DeepSeek-V3$_{671B}$	**0.7514**	0.3090	0.7846	0.6074	0.7389	0.4566	0.7039	0.4351	-	-
	DeepSeek-R1$_{671B}$	0.7491	0.1926	**0.7941**	0.3549	**0.8798**	0.1951	**0.7596**	0.2821	-	-
	Mistral$_{7B}$	0.6309	0.4714	0.5943	0.6175	0.5505	0.5209	0.3353	0.3127	-	-
Ret. Enh.	+Adaptive RAG	0.5782	0.3179	0.5999	0.5123	0.6235	0.2993	0.3615	0.3512	5.75	8.25
Ret. Enh.	+LongLLMLingua‡	0.5009	0.4049	0.4955	0.4961	0.3815	0.3786	0.2958	0.2765	9.50	8.25
Ret. Enh.	+LLMLingua-2	0.3522	0.0402	0.5166	0.1587	0.5405	0.1547	0.3142	0.0916	8.50	11.25
Ret. Enh.	+BGM	0.3484	0.3479	0.4102	0.4431	0.4645	0.4349	0.1513	0.1724	10.75	9.25
Generation Enh.	+ThoT‡	0.6290	0.4030	0.5989	0.5813	0.5835	0.4710	0.3887	0.3070	5.25	6.75
Generation Enh.	+SelfElicit‡	0.6271	0.4237	0.6110	0.5917	0.5410	0.5165	0.2249	0.1989	6.50	6.50
Generation Enh.	+MetaRAG‡	0.2268	0.2371	0.0383	0.0412	0.0350	0.0313	0.0192	0.0250	12.00	11.75
Generation Enh.	+Prompt Tuning	0.5970	0.5544	0.6387	0.6936	0.7220	0.7361	0.4976	0.5098	4.00	3.75
Generation Enh.	+Prefix Tuning	0.5085	0.4471	0.5646	0.6107	0.4690	0.4671	0.2651	0.2951	8.50	6.00
Generation Enh.	+SPRING	0.5951	0.5822	0.6665	0.7245	0.6145	0.5926	0.5122	0.5451	4.25	3.25
Generation Enh.	+R^{2}AG	0.6554	0.7015	0.6842	0.7557	0.7465	0.7665	0.5744	**0.6153**	1.75	1.75
Generation Enh.	+BubbleRAG (ours)	0.6893	**0.7323**	0.6958	**0.7600**	0.7520	**0.7705**	0.5547	0.6094	1.25	1.25

Table 1: Main results across four datasets. Enhanced RAG methods are categorized into: *Retrieval-enhanced RAG (Ret. Enh.)* and *Generation-enhanced RAG (Generation Enh.)*. † indicates a closed-source LLM, and ‡ means a training-free enhanced RAG method. The overall best result is marked in **bold**; the second-best is underlined. Results marked in deeper yellow point to higher scores with the same backbone LLM, and deeper blue gradients denote lower performance. Average ranking of enhanced RAG methods is also provided.

Dataset(→)	HotpotQA		2Wiki		MuSiQue	
Method(↓)/Metric(→)	Acc.	F1	Acc.	F1	Acc.	F1
BubbleRAG	0.6958	0.7600	0.7520	0.7705	0.5547	0.6094
w/o. thought bubble	0.6660 (↓4.3%)	0.7308 (↓3.8%)	0.6460 (↓14.1%)	0.6692 (↓13.1%)	0.5483 (↓1.2%)	0.5938 (↓2.6%)
w/o. interactive reading	0.6816 (↓2.0%)	0.7448 (↓2.0%)	0.6980 (↓7.2%)	0.7215 (↓6.4%)	0.5541 (↓0.1%)	0.5972 (↓2.0%)

Table 2: Ablation results of BubbleRAG with Mistral$_{7B}$. (%) indicates the relative performance gap.

4.4 Ablation Studies

To assess the contribution of the two modules in BubbleRAG, we conduct ablation studies with two variants. First, we exclude the thought bubble individually. Bookmark tokens are replaced with learnable tokens that do not encode the internal cognition from the LLM. Secondly, we replace the iterative marking process with a one-time process, where all bookmark tokens for each document are generated simultaneously by the thought bubble, rather than being produced in an interactive and autoregressive manner. The ablation studies are performed on the HotpotQA, 2Wiki, and MuSiQue datasets, using Mistral$_{7B}$ as the base LLM. The results shown in Table 2 indicate that BubbleRAG consistently outperforms all ablated variants, confirming the superiority of BubbleRAG. Notably, removing the thought bubble module leads to the most significant drop across all metrics, confirming that leveraging the LLM's intermediate reasoning state via the thought bubble is crucial for cognitive offloading. Furthermore, replacing the iterative reading process also negatively impacts performance, suggesting that iterative interaction enables deeper and more refined bookmark annotation.

4.5 Discussion

In this section, we focus on answering the following research questions: **RQ1**: Does BubbleRAG enhance LLMs' ability to identify and utilize documents? **RQ2**: How do bookmark tokens affect the attention patterns of LLMs?

4.5.1 Evidence Identification and Utilization

In LLMs, the attention layers are burdened with the task of evidence retrieval (Dong et al., 2025). To assess whether the LLM attends to the relevant documents, we convert the LLM's attention map into a document-level ranking signal following Wang et al. (2024a). Specifically, we extract the last token's attention outputs, average them across all

Dataset(→)	HotpotQA				2Wiki			
Method(↓)/Metric(→)	Acc@1	MAP@10	MRR@10	NDCG@10	Acc@1	MAP@10	MRR@10	NDCG@10
Contriever	0.7936	0.7426	0.8748	0.8442	**0.8195**	**0.7043**	**0.8905**	**0.8317**
Mistral_{7B}	0.5151	0.6784	0.7342	0.7862	0.5570	0.6419	0.7431	0.7713
+R^2AG	0.7735	0.7565	0.8684	0.8514	0.6580	0.7007	0.8087	0.8142
+BubbleRAG (ours)	**0.8002**	**0.7940**	**0.8829**	**0.8728**	0.7085	0.6650	0.8179	0.7967

Table 3: Attention-based retrieval results. The best and second-best results are marked in **bold** and underlined.

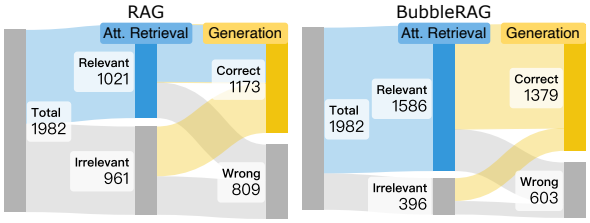

Figure 3: Results of RAG and BubbleRAG with Mistral_{7B} on HotpotQA, grouped by the highest attention-based relevance.

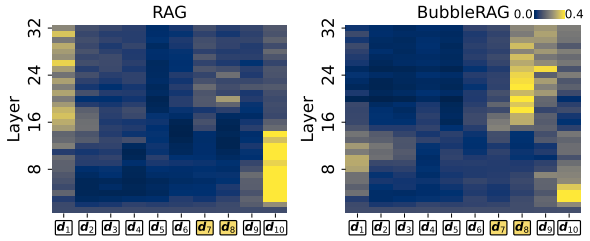

Figure 4: Document-level attention heatmaps of the last token in Mistral_{7B}, with the X-axis representing document indices and the Y-axis representing attention layers. Relevant documents are highlighted in yellow.

heads, and then compute the mean attention that each document's tokens receive. We regard the mean score as the document's ranking score, from which we derive retrieval metrics. Table 3 reports the attention-based retrieval results on HotpotQA and 2Wiki. It is clear that Mistral_{7B} obtains low Acc@1 scores, indicating that it often focuses on distracting content within the retrieved set. Even though R^2AG is explicitly trained with relevance labels, it still achieves moderate performance, showing that silent reading is insufficient to guide the LLM's attention toward the correct document. Conversely, BubbleRAG substantially increases attention to relevant documents, leading to significant improvements in retrieval metrics. On HotpotQA dataset, BubbleRAG achieves the best performance on all metrics. That suggests BubbleRAG enables LLMs to perform list-wise ranking and achieve evidence identification performance comparable to retrievers such as Contriever, which excels at capturing fine-grained distinctions (Zhu et al., 2024; Qorib et al., 2024). This complex retrieval ability is a key ingredient for LLMs to comprehend retrieved documents (Dong et al., 2025).

The LLM may pay greater attention to crucial documents but still produce incorrect answers (Liu et al., 2025). To validate whether BubbleRAG improves the utilization of attended evidence, we analyze the grouped generation performance on HotpotQA using Mistral_{7B}. Specifically, samples are divided into "relevant" and "irrelevant" groups, based on whether the most-attended document is relevant to the query. As shown in Figure 3, standard RAG produces many wrong answers even when attending to relevant evidence. In contrast, BubbleRAG significantly increases the proportion of correct answers when identifying relevant documents. These findings indicate that BubbleRAG not only guides the LLM to focus on the most informative document but also enables the LLM to utilize this document for answer generation.

4.5.2 Attention Visualization with Bookmarks

To better understand how bookmark tokens improve LLMs' generation, we present a visualization of the attention distribution in BubbleRAG compared to standard RAG, using Mistral_{7B} as the backbone LLM. Specifically, we focus on a case from the 2Wiki dataset, where the relevant documents are ranked in positions 7 and 8, with other documents serving as potential distractors.

We first visualize the attention distribution of the last token over document tokens, averaged across all attention heads. Figure 4 shows document-level attention heatmaps. It is evident that RAG tends to disproportionately focus on an irrelevant document positioned at the end of the ranking, neglecting the more relevant middle documents. This bias potentially misguides the LLM, leading to erroneous responses (Liu et al., 2023). In contrast, BubbleRAG directs more attention to the relevant documents, especially in the deeper layers (16-32),

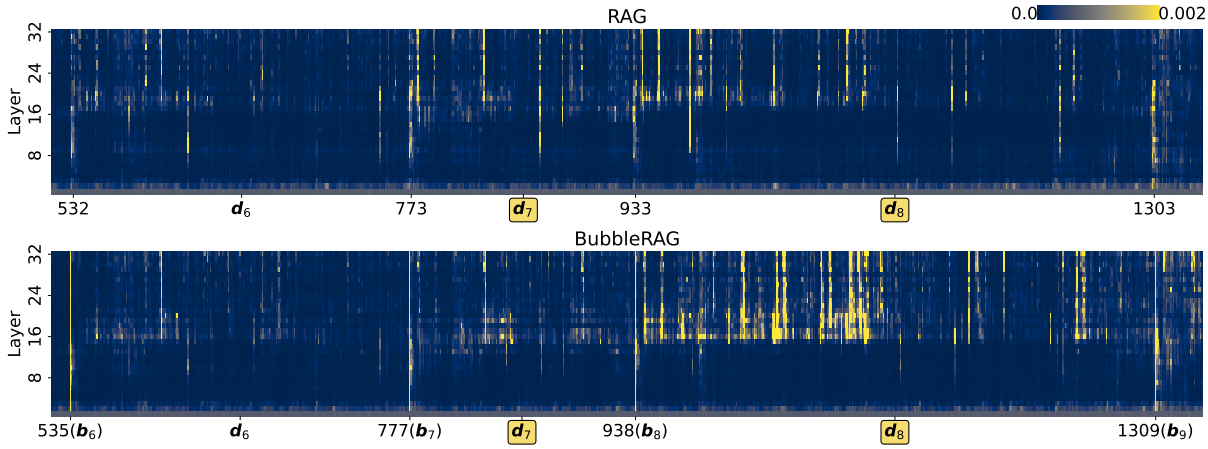

Figure 5: Heatmaps of token-level attention distribution for the last token using Mistral_{7B}, with the X-axis representing tokens and the Y-axis representing layers.

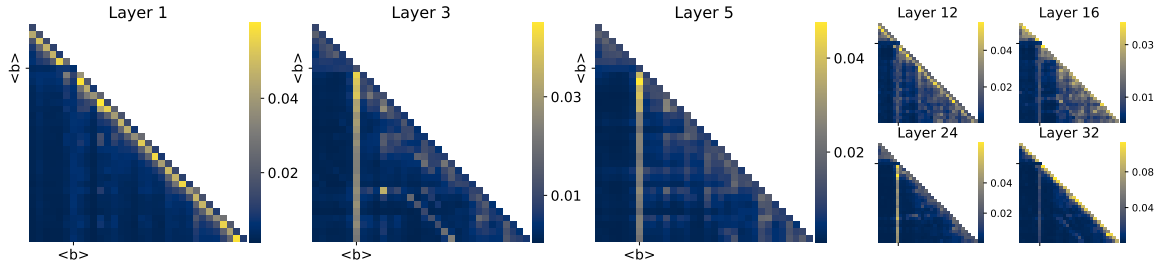

Figure 6: Average attention patterns for 32 tokens near the bookmark tokens across 10 candidate documents using Mistral_{7B}. The first bookmark token is labeled as "<b>".

effectively identifies the irrelevant documents, and steers the LLM's focus toward more useful information. While the last document initially receives relatively high attention before the 16th layer, the LLM gradually deprioritizes as it is recognized as irrelevant (Ju et al., 2024). Overall, relevant documents get more attention, consistent with the attention-based retrieval analysis in Discussion 4.5.1.

Next, at the token level, as shown in Figure 5, we analyze the attention distribution for tokens within the top-6 to top-8 ranked documents. It is obvious that the bookmark tokens receive higher attention scores even in deeper layers. That suggests bookmark tokens effectively act as cognitive links, bridging documents and effectively guiding the LLM's attention toward valuable information.

Figure 6 further illustrates attention patterns near the bookmark tokens. We can draw several conclusions: (1) In the 1st layer, attention is more focused on recent tokens, displaying a local, sparse pattern as discussed in Xiao et al. (2024). (2) Between the 3rd and 24th layers, a clear pattern emerges where the information flow around the bookmark tokens converges, forming an arrow-shaped attention pattern like attention sinks (Xiao et al., 2024; Barbero et al., 2025). This convergence pattern aligns with the observations of "anchor token" (Wang et al., 2023), where certain tokens act as reference points for the LLM's decision-making process. (3) In the 32nd layer, the LLM shifts its focus back to recent tokens, exhibiting a localized attention pattern (Wan et al., 2025).

See Appendix B for additional discussion and extended experiments.

5 Conclusion

In this work, we investigate limitations of the silent reading paradigm in RAG frameworks, particularly their inability to offload and revisit internal cognition when integrating retrieved evidence. To address this challenge, we propose BubbleRAG, an enhanced RAG method that inserts bookmarks via an interactive reading process using a thought bubble module. By externalizing the LLM's internal cognition and inserting it back into documents, BubbleRAG improves evidence integration and answer quality. Extensive experiments demonstrate that BubbleRAG outperforms baseline RAG methods and enables LLMs to better focus on and utilize relevant documents. Future work will explore more flexible approaches for integrating bookmarks.

Limitations

BubbleRAG has several limitations. First, BubbleRAG relies on access to intermediate hidden states of the backbone LLM, which may not be available in some closed-source or API-only models. Secondly, the interactive reading process introduces additional computational overhead due to bookmark generation. Third, the generated bookmark tokens are primarily designed for LLM consumption and are not directly interpretable by humans, although our analyses show they are meaningful and effective for model reasoning. Finally, BubbleRAG focuses on textual QA tasks; the effectiveness in other settings, such as reasoning generation or multi-modal QA, remains to be explored.

Ethics Statement

LLMs may produce inaccurate or misleading outputs, which can lead to potential risks in real-world applications. BubbleRAG is designed to mitigate this issue by encouraging more faithful evidence identification and utilization through interactive reading and externalized cognition, rather than relying solely on internal reasoning. Our work adheres to established ethical standards in the research community. All datasets and models used in this study are publicly available and were employed in accordance with their intended use and licenses. We do not collect or use any personal or sensitive data, and the authors declare no conflicts of interest.

Acknowledgments

Lei Chen is supported by National Key Research and Development Program of China Grant No. 2023YFF0725100, National Science Foundation of China under Grant No. U22B2060, Guangdong-Hong Kong Technology Innovation Joint Funding Scheme Project No. 2024A0505040012, AOE Project AoE/E-603/18, Theme-based project TRS T41-603/20R, CRF Project C2004-21G, Key Areas Special Project of Guangdong Provincial Universities 2024ZDZX1006, Guangdong Province Science and Technology Plan Project 2023A0505030011, HKUST(GZ) CMCC(Guangzhou Branch) Metaverse Joint Innovation Lab under Grant No. P00659, Hong Kong ITC TC-SKLCRCC26EG01, ITF grant PRP/004/22FX, Zhujiang scholar program 2021JC02X170, HKUST Webank joint research lab. Jiachuan Wang's work is supported in part by JST CREST (JPMJCR22M2). Yongqi Zhang's work is supported by Guangdong Provincial Natural Science Foundation 2025A1515010304, Guangdong Province Project 2024QN11X088, Guangzhou Science and Technology Planning Project 2025A03J4491. Shuangyin Li's work is supported by National Natural Science Foundation of China No. 62006083.

References

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, and 260 others. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

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

Ingeol Baek, Hwan Chang, ByeongJeong Kim, Jimin Lee, and Hwanhee Lee. 2025. Probing-RAG: Self-probing to guide language models in selective document retrieval. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3287–3304, Albuquerque, New Mexico. Association for Computational Linguistics.

Jiaqi Bai, Hongcheng Guo, Jiaheng Liu, Jian Yang, Jian Yang, Xinnian Liang, Zhao Yan, and Zhoujun Li. 2023. Griprank: Bridging the gap between retrieval and generation via the generative knowledge improved passage ranking. *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*.

Federico Barbero, Alvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Petar Veličković, Razvan Pascanu, and Michael M. Bronstein. 2025. Why do LLMs attend to the first token? In *Second Conference on Language Modeling*.

Tianchi Cai, Zhiwen Tan, Xierui Song, Tao Sun, Jiyan Jiang, Yunqi Xu, Yinger Zhang, and Jinjie Gu. 2024. Forag: Factuality-optimized retrieval augmented generation for web-enhanced long-form question answering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 199–210, New York, NY, USA. Association for Computing Machinery.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Andy Clark. 2010. *Supersizing the mind: Embodiment, action, and cognitive extension*. oxford university Press.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 179 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv*, abs/2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bing-Li Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dong-Li Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 179 others. 2024. Deepseek-v3 technical report. *ArXiv*, abs/2412.19437.

Yihe Dong, Lorenzo Noci, Mikhail Khodak, and Mufan Li. 2025. Attention retrieves, mlp memorizes: Disentangling trainable components in the transformer. *ArXiv*, abs/2506.01115.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 510 others. 2024. The llama 3 herd of models. *ArXiv*, abs/2407.21783.

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

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *ArXiv*, abs/2312.10997.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning LLMs on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7765–7784, Miami, Florida, USA. Association for Computational Linguistics.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715, Seattle, United States. Association for Computational Linguistics.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938.

Kyubeen Han, Junseo Jang, Hongjin Kim, Geunyeong Jeong, and Harksoo Kim. 2025. Exploring the impact of instruction-tuning on LLM's susceptibility to misinformation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26711–26731, Vienna, Austria. Association for Computational Linguistics.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason E Weston, and Yuandong Tian. 2025. Training large language models to reason in a continuous latent space. In *Workshop on Reasoning and Planning for Large Language Models*.

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.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong 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, Mexico City, Mexico. Association for Computational Linguistics.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O Arik. 2025. Long-context LLMs meet RAG: Overcoming challenges for long inputs in RAG. In *International Conference on Learning Representations*.

Martin Johnson and Stuart Shaw. 2008. Annotating to comprehend: A marginalised activity?

Tianjie Ju, Weiwei Sun, Wei Du, Xinwei Yuan, Zhaochun Ren, and Gongshen Liu. 2024. How large language models encode context knowledge? a layer-wise probing study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8235–8246.

Zixuan Ke, Weize Kong, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Bridging the preference gap between retrievers and LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10438–10451, Bangkok, Thailand. Association for Computational Linguistics.

Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. 2023. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 996–1009, Singapore. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

David Kirsh. 2010. Thinking with external representations. *AI Soc.*, 25(4):441–454.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc V. Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *ArXiv*, abs/2203.05115.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*.

Bo Li, Tian Tian, Zhenghua Xu, Hao Cheng, Shikun Zhang, and Wei Ye. 2026a. Modeling uncertainty trends for timely retrieval in dynamic RAG. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 31527–31535. AAAI Press.

Bo Li, Mingda Wang, Gexiang Fang, Shikun Zhang, and Wei Ye. 2026b. Retrieval as generation: A unified framework with self-triggered information planning. *ArXiv*, abs/2604.11407.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Zhicheng Liu and John T. Stasko. 2010. Mental models, visual reasoning and interaction in information visualization: A top-down perspective. *IEEE Transactions on Visualization and Computer Graphics*, 16:999–1008.

Zhining Liu, Rana Ali Amjad, Ravinarayana Adkathimar, Tianxin Wei, and Hanghang Tong. 2025. SelfElicit: Your language model secretly knows where is the relevant evidence. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9153–9173, Vienna, Austria. Association for Computational Linguistics.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981, Bangkok, Thailand. Association for Computational Linguistics.

Muhammad Reza Qorib, Geonsik Moon, and Hwee Tou Ng. 2024. Are decoder-only language models better than encoder-only language models in understanding word meaning? In *Findings of the Association for Computational Linguistics ACL 2024*, pages 16339–16347.

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

Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, and et al. 2025. Investigating the factual knowledge boundary of large language models with retrieval augmentation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3697–3715.

Evan F. Risko and Sam J. Gilbert. 2016. Cognitive offloading. *Trends in Cognitive Sciences*, 20(9):676–688.

Devendra Singh Sachan, Siva Reddy, William L. Hamilton, Chris Dyer, and Dani Yogatama. 2021. End-to-end training of multi-document reader and retriever for open-domain question answering. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Sohel Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *ArXiv*, abs/2402.07927.

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.

Tianyuan Shi, Liangzhi Li, Zijian Lin, Tao Yang, Xiaojun Quan, and Qifan Wang. 2023. Dual-feedback knowledge retrieval for task-oriented dialogue systems. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6566–6580, Singapore. Association for Computational Linguistics.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-augmented black-box language models. 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 8371–8384, Mexico City, Mexico. Association for Computational Linguistics.

John Sweller, Jeroen J. G. van Merriënboer, and Fred Paas. 2019. Cognitive architecture and instructional design: 20 years later. *Educational Psychology Review*, 31:261 – 292.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

H. Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2021. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Zhongwei Wan, Xinjian Wu, Yu Zhang, Yi Xin, Chaofan Tao, Zhihong Zhu, Xin Wang, Siqi Luo, Jing Xiong, Longyue Wang, and Mi Zhang. 2025. Dynamic discriminative operations for efficient generative inference of LLMs. In *International Conference on Learning Representations*.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label words are anchors: An information flow perspective

for understanding in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855, Singapore. Association for Computational Linguistics.

Tevin Wang, Jingyuan He, and Chenyan Xiong. 2024a. RAGViz: Diagnose and visualize retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 320–327, Miami, Florida, USA. Association for Computational Linguistics.

Zihao Wang, Anji Liu, Haowei Lin, Jiaqi Li, Xiaojian Ma, and Yitao Liang. 2024b. Rat: Retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation. *ArXiv*, abs/2403.05313.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*.

Yuan Xia, Jingbo Zhou, Zhenhui Shi, Jun Chen, and Hai ting Huang. 2024. Improving retrieval augmented language model with self-reasoning. In *AAAI Conference on Artificial Intelligence*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2023. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *International Conference on Learning Representations*.

Shicheng Xu, Liang Pang, Jun Xu, Huawei Shen, and Xueqi Cheng. 2024. List-aware reranking-truncation joint model for search and retrieval-augmented generation. In *Proceedings of the ACM on Web Conference 2024*, pages 1330–1340.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *ArXiv*, abs/2401.15884.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *ArXiv*, abs/2505.09388.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Fuda Ye, Shuangyin Li, Yongqi Zhang, and Lei Chen. 2024. R^2AG: Incorporating retrieval information into retrieval augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11584–11596.

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

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025. Reasoning models know when they're right: Probing hidden states for self-verification. In *Second Conference on Language Modeling*.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. *ArXiv*, abs/2402.19473.

Wenzhuo Zhao and Shuangyin Li. 2025. RUBY: An effective framework for multi-constraint multi-hop question generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18164–18188, Vienna, Austria. Association for Computational Linguistics.

Yucheng Zhou, Xiubo Geng, Tao Shen, Chongyang Tao, Guodong Long, Jian-Guang Lou, and Jianbing Shen. 2023. Thread of thought unraveling chaotic contexts. *ArXiv*, abs/2311.08734.

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

Yilun Zhu, Joel Ruben Antony Moniz, Shruti Bhargava, Jiarui Lu, and et al. 2024. Can large language models understand context? In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 2004–2018.

Yutao Zhu, Zhaoheng Huang, Zhicheng Dou, and Ji-Rong Wen. 2025a. One token can help! learning scalable and pluggable virtual tokens for retrieval-augmented large language models. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2025b. Large language models for information retrieval: A survey. *ACM Trans. Inf. Syst.*, 44(1).

Appendix

This appendix is organized as follows:

- Section A is the algorithm of BubbleRAG.
- Section B provides further discussion.
- Section C presents extended experiments, including extended main results and extended ablation studies.
- Section D.1 provides dataset statistics.
- Section D.2 presents expanded baseline implementation details.
- Section D.3 lists the prompt templates.

A BubbleRAG Algorithm

Algorithm 1 is the pseudocode of BubbleRAG.

Algorithm 1: BubbleRAG

Input : Query $\mathbf{q}$, top-k retrieved documents $\mathbf{D}=\{\mathbf{d}_i\}_{i=1}^{k}$.
Output : Predicted answer $\hat{a}$.

Initialize LLM $\mathcal{G}(\cdot)$, thought bubble $f_{\text{bubble}}(\cdot)$.
Initialize annotated documents $\mathbf{D}^\star \leftarrow \emptyset$.
for $i = 1$ *to* k **do**
  Construct prompt $\mathbf{P}_i^\star$; // {Eq. 5}
  Compute hidden state $\mathbf{h}_i$; // KV caching is used during inference. {Eq. 6}
  Generate a bookmark token $\mathbf{b}_i$; // {Eq. 7}
  Append bookmark $\mathbf{d}_i^\star \leftarrow \mathbf{d}_i \oplus \mathbf{b}_i$; // {Eq. 8}
  Append $\mathbf{d}_i^\star$ to $\mathbf{D}^\star$;
end
Construct final prompt $\mathbf{P}^\star$ and generate answer $\hat{a}$; // {Eq. 9}
// Training objective
if *is training* **then**
  Calculate $\mathcal{L}_{\text{LM}}$; // {Eq. 10}
  Update parameters.
end
return $\hat{a}$

B Further Discussion

In this section, we further explore the following research questions. **RQ3**: How well does BubbleRAG generalize to varying candidate sizes? **RQ4**: How does the prompt template affect results?

B.1 RQ3: Generalizability to Varying Candidate Sizes

We examine the robustness of BubbleRAG to varying numbers of retrieved documents by training the model with $k = 10$ and evaluating its generalization to other candidate set sizes. Figure 7 presents the comparative performance of standard RAG and BubbleRAG using Mistral$_{7B}$ as the base LLM as the number of input documents changes. The results demonstrate several trends. First, BubbleRAG consistently outperforms standard RAG across all document counts on both NQ-30, HotpotQA, and 2Wiki datasets. Both RAG and BubbleRAG show clear improvements in performance when the number of documents increases on HotpotQA and 2Wiki datasets. On the NQ-30 dataset, BubbleRAG's performance also improves as the candidate set size grows up to $k = 10$, after which it slightly declines. This trend may indicate a trade-off: while a moderate number of documents introduces more relevant evidence, extensive candidate sets increase noise and distract the model, mildly affecting performance (Jin et al., 2025). Importantly, BubbleRAG's performance remains strong even when the evaluation k does not precisely match the training setting, highlighting the model's robustness to dynamic retrieval environments.

B.2 RQ4: Impact of Prompt Template

In our main experiments, the prompt template employed a setting where the question was placed both before and after the set of documents. In this section, we evaluate model performance under an alternative template where the question appears only after the documents. This revised prompt configuration is consistent with the template described in Section D.3, except that the initial occurrence of the question is omitted. Table 4 summarizes the results. The findings indicate that this simplified prompt template leads to a consistent performance degradation across most methods and datasets, suggesting that repeating the question helps reinforce task relevance during generation. Nevertheless, BubbleRAG remains the strongest performer under this setting, achieving the near-best results across all benchmarks. Importantly, the overall conclusions regarding the relative effectiveness of BubbleRAG and competing methods remain consistent with those drawn from the main results.

C Extended Experiments on LLaMA2$_{7B}$ and Qwen3$_{8B}$

In this section, we conduct extended experiments with two popular open-source LLMs: LLaMA2$_{7B}$ (Touvron et al., 2023) and Qwen3$_{8B}$ (Yang et al., 2025), which represent weaker and stronger LLMs, respectively, within a similar parameter.

C.1 Extended Main Results

Table 5 reports the extended results on LLaMA2$_{7B}$ and Qwen3$_{8B}$. Several observations can be drawn:

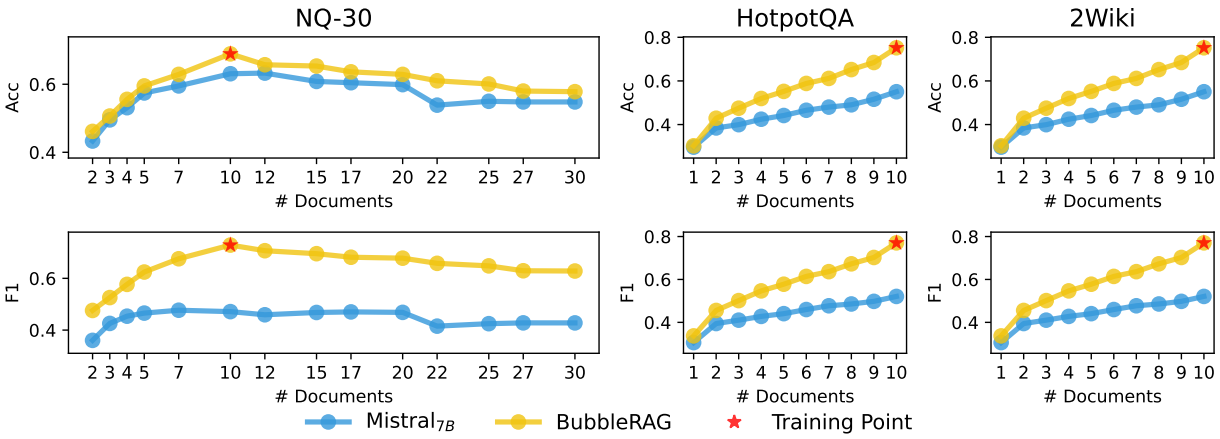

Figure 7: Performance of RAG and BubbleRAG across different document counts. Training point means its k aligns with the training setting.

Dataset(→)	NQ-10		HotpotQA		2Wiki		MuSiQue	
Method(↓)/Metric(→)	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Mistral_{7B}	0.6083	0.4451	0.5787	0.5691	0.4745	0.4173	0.3418	0.2993
+Adaptive RAG	0.5782	0.3644	0.5696	0.4674	0.5585	0.3133	0.3402	0.2977
+LLMLingua-2	0.3522	0.0402	0.5166	0.1587	0.5405	0.1547	0.3142	0.0916
+BGM	0.3484	0.3479	0.4102	0.4431	0.4645	0.4349	0.1513	0.1724
+Prompt Tuning	0.5989	0.6273	0.6831	**0.7654**	0.6750	0.6816	0.5683	0.6142
+Prefix Tuning	0.4934	0.4704	0.5691	0.6282	0.4655	0.4544	0.2881	0.3392
+SPRING	0.5650	0.5822	0.6665	0.7245	0.6145	0.5926	0.5122	0.5451
+R^2AG	0.6215	0.6790	0.6796	0.7617	0.7190	0.7338	0.5596	0.6072
+BubbleRAG (ours)	**0.6987**	**0.7458**	**0.6842**	0.7617	**0.7220**	**0.7362**	**0.5758**	**0.6234**

Table 4: Main results across four datasets under the prompt template where the question appears only after retrieved documents.

(1) BubbleRAG consistently improves performance across model capacities, while stronger LLMs benefit more from its interactive reading mechanism. Although LLaMA2_{7B} is intrinsically weaker than Qwen3_{8B}, BubbleRAG significantly improves its performance on all datasets. Notably, on the 2Wiki dataset, LLaMA2_{7B} with BubbleRAG surpasses Qwen3_{8B} using standard RAG.

(2) Reasoning-based methods show limited effectiveness for weaker LLMs. Training-free reasoning approaches such as ThoT and Self-Elicit fail to consistently improve LLaMA2_{7B} and often degrade F1. Although these methods yield moderate gains on Qwen3_{8B}, their improvements remain unstable and inferior to those achieved by BubbleRAG.

C.2 Extended Ablation Studies

The ablation results with LLaMA2_{7B} and Qwen3_{8B} are provided in Table 6. As shown, removing any single component — thought bubble or iterative reading — consistently leads to a drop in both accuracy and F1 scores across all evaluated datasets.

To further address whether BubbleRAG's gains can be explained by explicit natural-language compression alone, we add a budget-aligned textual bookmark ablation on NQ-10 and HotpotQA with Mistral_{7B}. Specifically, at each reading step, we prompt the LLM to generate a fixed-length natural-language annotation for each retrieved document and append the annotation to the document before answer generation. Because enforcing a strict one-token-per-document textual control is impractical, we use 32 annotation tokens per document. To make this baseline stronger, we also test a variant where the middle-step annotations are generated by Qwen3_{8B}.

As shown in Table 7, textual bookmarks provide only limited gains, even when annotations are generated by a stronger LLM. In contrast, BubbleRAG remains substantially better on both datasets. Notably, this textual control uses a much larger budget (about 32× per-document annotation cost) than

Dataset(→)	NQ-10		HotpotQA		2Wiki		MuSiQue		Avg. Ranking	
Method(↓)/Metric(→)	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
LLaMA2$_{7B}$	0.4765	0.3071	0.4440	0.3923	0.4110	0.3256	0.1935	0.1539	-	-
+Adaptive RAG	0.4501	0.2314	0.4536	0.3190	0.4235	0.1520	0.1908	0.1491	4.75	7.25
+LongLLMLingua‡	0.4105	0.2193	0.4198	0.3220	0.3590	0.2737	0.2074	0.1420	6.00	7.25
+LLMLingua-2	0.1638	0.0382	0.3088	0.1715	0.3375	0.0924	0.0812	0.0389	8.75	10.75
+BGM	0.2599	0.2197	0.3017	0.3315	0.4410	0.4048	0.1227	0.1258	7.50	6.50
+ThoT‡	0.1431	0.0788	0.2185	0.1403	0.2345	0.1474	0.0564	0.0541	10.75	10.25
+SelfElicit‡	0.4501	0.2788	0.4415	0.3474	0.4070	0.2240	0.1792	0.1413	5.50	6.50
+Prompt Tuning	0.4218	0.4672	0.5908	0.6532	0.5320	0.3606	0.3573	0.4269	3.75	3.25
+Prefix Tuning	0.1092	0.1643	0.3047	0.3726	0.2700	0.2932	0.0768	0.1324	10.00	7.00
+SPRING	0.4670	0.3417	0.4162	0.4156	0.3420	0.3178	0.2036	0.1686	5.75	4.25
+R^{2}AG	0.5556	0.5851	0.6034	0.6748	0.6265	0.6426	0.4085	0.4658	2.00	2.00
+BubbleRAG (ours)	0.6196	0.6688	0.6302	0.6962	0.6720	0.6882	0.4559	0.5037	1.00	1.00
Qwen3$_{8B}$	0.7137	0.2629	0.7089	0.5751	0.6685	0.4705	0.4604	0.3053	-	-
+Adaptive RAG	0.6723	0.2359	0.6847	0.5285	0.6700	0.3760	0.5186	0.3833	5.00	8.00
+LongLLMLingua‡	0.5989	0.2054	0.5782	0.4597	0.4670	0.3643	0.4491	0.3171	8.00	9.25
+LLMLingua-2	0.1205	0.0201	0.3920	0.0934	0.3960	0.1047	0.1826	0.0486	10.75	10.75
+BGM	0.3465	0.1713	0.4157	0.3311	0.3140	0.2682	0.1987	0.1446	12.00	12.00
+ThoT‡	0.7100	0.2619	**0.7245**	0.5451	0.7230	0.3814	0.5433	0.4054	2.75	7.00
+SelfElicit‡	0.3126	0.1802	0.5610	0.4986	0.0240	0.0159	0.2068	0.1696	10.25	10.00
+MetaRAG‡	0.5650	0.5860	0.6791	0.7618	**0.7535**	0.7253	0.4148	0.4556	6.00	4.25
+Prompt Tuning	0.5763	0.6416	0.7038	0.7798	0.6880	0.7033	0.5780	0.6262	4.75	3.25
+Prefix Tuning	0.5443	0.4777	0.6549	0.7285	0.5640	0.5751	0.4730	0.5355	7.75	5.00
+SPRING	0.6855	0.4645	0.6635	0.7193	0.6350	0.6229	0.4529	0.4772	6.00	5.50
+R^{2}AG	0.6610	0.7312	0.7069	0.7917	0.7335	0.7465	0.5855	0.6300	3.25	2.00
+BubbleRAG (ours)	**0.7326**	**0.7769**	0.7200	**0.7933**	0.7475	**0.7655**	**0.5952**	**0.6317**	1.50	1.00

Table 5: Main results across four datasets using LLaMA2$_{7B}$ and Qwen3$_{8B}$.

Dataset(→)	HotpotQA		2Wiki		MuSiQue	
Method(↓)/Metric(→)	Acc.	F1	Acc.	F1	Acc.	F1
Qwen3$_{8B}$+BubbleRAG	0.7200	0.7933	0.7475	0.7655	0.5952	0.6317
w/o. thought bubble	0.6983 (↓3.0%)	0.7681 (↓3.2%)	0.6660 (↓10.9%)	0.6873 (↓10.2%)	0.5697 (↓4.3%)	0.6138 (↓2.8%)
w/o. interactive reading	0.7094 (↓1.5%)	0.7831 (↓1.3%)	0.7135 (↓4.5%)	0.7309 (↓4.5%)	0.5753 (↓3.3%)	0.6289 (↓0.4%)
LLaMA2$_{7B}$+BubbleRAG	0.6302	0.6962	0.6720	0.6882	0.4559	0.5037
w/o. thought bubble	0.5752 (↓8.7%)	0.6334 (↓9.0%)	0.5485 (↓18.4%)	0.5617 (↓18.4%)	0.3746 (↓17.8%)	0.4380 (↓13.0%)
w/o. interactive reading	0.6317 (↑0.2%)	0.7109 (↑2.1%)	0.6595 (↓1.9%)	0.6793 (↓1.3%)	0.4365 (↓4.3%)	0.4873 (↓3.3%)

Table 6: Ablation results of BubbleRAG with LLaMA2$_{7B}$ and Qwen3$_{8B}$. (%) indicates the relative performance gap compared to the BubbleRAG.

BubbleRAG's implicit one-bookmark mechanism, yet still cannot match BubbleRAG's performance. These results suggest that BubbleRAG's advantage is not merely due to adding more text tokens, but comes from its interactive and compact cognitive offloading design.

D Experiment Details

D.1 Dataset Details

Natural Questions (NQ) (Kwiatkowski et al., 2019) pairs real Google search queries with human-annotated answers from Wikipedia. We use the version standardized by (Liu et al., 2023), where each query is accompanied by $k-1$ distractor documents retrieved by Contriever (Izacard et al., 2021). We use the 10-candidate version (NQ-10) in the experiments. We also utilize the 30-candidate version (NQ-30) in the discussion B.1.

HotpotQA (Yang et al., 2018) is a widely used multi-hop question answering dataset that requires synthesizing evidence from multiple Wikipedia articles. Under the distractor setting, each question pairs with 10 documents, and queries are categorized as either "bridging" or "comparison" according to the required reasoning pattern.

2WikiMultiHopQA (2Wiki) (Ho et al., 2020) consists of questions involving up to 5 reasoning hops, each paired with 10 documents. Unlike Hot-

Dataset(→)	NQ-10		HotpotQA	
Method(↓)/Metric(→)	Acc.	F1	Acc.	F1
Mistral_{7B}	0.6309	0.4714	0.5943	0.6175
+Textual Bookmarks	0.6271	0.4832	0.5918	0.6173
+Textual Bookmarks (Qwen3)	0.6365	0.4962	0.5974	0.6206
+BubbleRAG (ours)	**0.6893**	**0.7323**	**0.6958**	**0.7600**

Table 7: Budget-aligned textual bookmark ablation compared with BubbleRAG.

potQA, 2Wiki evaluates models not only based on supporting evidence but also their ability to identify relevant entity-relation tuples.

MuSiQue (Trivedi et al., 2021) contains questions that require 2–4 reasoning hops across six predefined patterns. The dataset is constructed using a bottom–up process by carefully selecting and composing single-hop questions. In the distractor split, the correct answer must be identified among 20 candidate documents.

Dataset statistics are reported in Table 8.

D.2 Baseline Details

LLMs (Standard RAG)

GPT-4o (Ouyang et al., 2022; Achiam et al., 2023) are popular closed-source LLMs that we access through the OpenAI API. The version of GPT-4o is "gpt-4o-2024-11-20". The temperature is fixed at 0.0.

DeepSeek-V3 and DeepSeek-R1 (DeepSeek-AI et al., 2024, 2025) are open-source 671B LLMs that achieve impressive performance across various benchmarks. We invoke the versions "deepseek-chat" ("DeepSeek-V3-0324") and "deepseek-reasoner" ("DeepSeek-R1") via the DeepSeek API platform[1] with the temperature set to 0.0. As DeepSeek-R1 outputs an explicit reasoning trace before its final answer, we evaluate only the segment that follows the "<\think>" delimiter.

LLaMA2 (Touvron et al., 2023) is a popular open-source family of LLMs. Our experiments use two versions: "Llama-2-7b-chat-hf" and "Llama-2-13b-chat-hf". We adopt greedy decoding to ensure a fair comparison. The length and repetition penalty are both set to 1.0, while the temperature is set to 1.0. All other settings are set to their default values.

Mistral (Jiang et al., 2023) is another competitive open-source LLM. In our experiments, we use the "Mistral-7B-Instruct-v0.3" version. We apply the same decoding hyper-parameters as for LLaMA2_{7B}.

Qwen3 (Yang et al., 2025) is the latest series of open-source LLMs. We use the "Qwen3-8B" version. Consistent with our setup for other LLMs, we apply greedy decoding with the same hyper-parameters as used for LLaMA2_{7B}.

Under these configurations, LLaMA2_{7B}, Mistral_{7B}, and Qwen3_{8B} serve as the base LLMs for all enhanced RAG baselines in our experiments.

Retrieval Enhancement

Adaptive RAG (Jeong et al., 2024) augments the standard RAG pipeline with a classifier that chooses, per query, whether the LLM should consult external evidence. To keep the comparison fair, we restrict the action space to just two options: "non-retrieval" and "retrieval". A query is labeled "non-retrieval" if the base LLM can directly produce the correct answer without retrieved documents; all others receive the "retrieval" label. The action is predicted by a T5 ("flan-t5-base") (Chung et al., 2024) classifier fine-tuned on these labels.

LongLLMLingua (Jiang et al., 2024) compresses an input prompt by exploiting the information–salient distribution predicted by a smaller LLM. The compression process controls the token budget while retaining critical content. Following the authors' recommendations, we employ LLaMA2_{7B} "Llama-2-7b-chat-hf" (Touvron et al., 2023) as the compressor and fix the compression rate at 0.55.

LLMLingua-2 (Pan et al., 2024) formulates prompt compression as a token-classification task: an encoder scores each token for retention or deletion. Following previous work, we fine-tune "xlm-roberta-large" (Conneau et al., 2020) as the token classification model on each dataset, leaving all hyper-parameters at their default settings.

BGM (Ke et al., 2024) trains a bridging model via supervised learning, followed by reinforcement

[1] https://platform.deepseek.com

Datasets	# Query	# Train/Test	# Tokens	# Rel/Docs	MAP	MRR@10	NDCG@10
NQ-10	2655	2124/531	~2k	1/10	0.5170	0.5170	0.6288
NQ-30	2655	2124/531	~6k	1/30	0.5170	0.5170	0.6288
HotpotQA	9906	7924/1982	~2k	2.38/10	0.7426	0.8748	0.8442
2Wiki	10000	8000/2000	~2k	2.44/10	0.7043	0.8905	0.8317
MuSiQue	9856	7894/1962	~3k	2.10/20	0.6194	0.8423	0.7127

Table 8: Statistics of datasets. "# Rel/Docs" denotes the number of relevant documents and the total number of documents for each query. "MAP", "MRR@10", and "NDCG@10" are common retrieval metrics.

learning to select the documents most preferred by the target LLM. The bridging model outputs indices over the retrieved materials, and the corresponding documents constitute the final evidence sent to the LLM. Because the authors' T5 implementation is limited to a 512-token context window, we substitute LLaMA3.2$_{1B}$ ("Llama-3.2-1B-Instruct") (Dubey et al., 2024), whose window we extend to 8k tokens. For stable generation, the bridging model's output space is restricted to k reserved special tokens, each representing a candidate document.

Generation Enhancement

ThoT (Zhou et al., 2023) prompts LLMs to handle noisy contexts by examining each document step by step and summarizing the key findings throughout the process. To prevent potential information leakage from intermediate reasoning, we discard the reasoning traces and evaluate the method solely based on its final output.

SelfElicit (Liu et al., 2025) is an inference-time augmentation that taps into the LLM's own attention maps to locate the most salient evidence sentences in the supplied context and then explicitly highlights them before the final generation step. Following the authors' recommended setup, we keep all other hyper-parameters at their defaults to generate final answers.

MetaRAG (Zhou et al., 2024) is a metacognitive framework that enhances RAG systems by enabling the LLM to monitor, evaluate, and regulate its own retrieval and reasoning processes. In our experiments, we follow the authors' open-source implementation and evaluate the performance based on its final generated response. For a fair comparison with non-iterative baselines, we constrain the monitoring, evaluation, and planning stages of MetaRAG to operate over only the top-k retrieved candidates.

Prompt Tuning (Lester et al., 2021) learns a task-specific sequence of trainable prompt tokens that is prepended to the original input, guiding the LLM toward the desired behavior without altering its parameters. For all experiments, we set the number of learnable tokens to k to ensure a fair comparison across baselines.

SPRING (Zhu et al., 2025a) appends a set of learnable virtual tokens that are simply added to the retrieved documents and user input. The number of virtual tokens is set to 50, and these tokens will be trained using the same supervised language-modeling target as used for BubbleRAG on the specific dataset.

Prefix Tuning (Li and Liang, 2021) prepends a sequence of continuous prefix vectors to every attention layer, allowing the frozen backbone LLM to adapt with minimal overhead. In our implementation, we prepend k visual tokens produced by a lightweight prefix encoder composed of a single embedding layer.

R^{2}AG (Ye et al., 2024) computes a suite of retrieval-level features based on semantic similarities. These features are injected into the LLM through an auxiliary R^2-Former, which aligns the retrieval signal with the LLM's input space. We follow the authors' training strategy and fine-tune the R^2-Former for each dataset. The attention layer's head number and hidden size in R^2-Former are 4 and 256, respectively.

D.3 Prompt Templates

The prompt templates in our experiments are summarized in Table 9. Following Liu et al. (2023), we utilize a query-aware contextualization template (placing the query before and after the documents) to achieve optimal performance for all RAG methods. To ensure a more complete evaluation, we also report results using the standard prompt template, where the query appears only after the documents. The corresponding results for this setting are presented in Appendix B.2.

Method	Prompt Template
RAG	Write a high-quality answer for the given question using only the provided search results (some of which might be irrelevant). Only give me the answer and do not output any other words. Question: {#**q**} [1]{#$\mathbf{d}_1$} [2]{#$\mathbf{d}_2$} ... [k]{#$\mathbf{d}_k$} Only give me the answer and do not output any other words. Question: {#**q**} Answer:
ThoT	Write a high-quality answer for the given question using only the provided search results (some of which might be irrelevant). Only give me the answer and do not output any other words. Question: {#**q**} [1]{#$\mathbf{d}_1$} [2]{#$\mathbf{d}_2$} ... [k]{#$\mathbf{d}_k$} Only give me the answer and do not output any other words. Question: {#**q**} Walk me through this context in manageable parts step by step, summarizing and analyzing as we go. {#*Reasoning process*} Therefore, the answer:
Comp.	Write a high-quality answer for the given question using only the provided search results (some of which might be irrelevant). Only give me the answer and do not output any other words. Question: {#**q**} {#*Compressed documents*} Only give me the answer and do not output any other words. Question: {#**q**} Answer:
R^2AG	Write a high-quality answer for the given question using only the provided search results (some of which might be irrelevant). Only give me the answer and do not output any other words. Question: {#**q**} [1]<s>{#$\mathbf{d}_1$} [2]<s>{#$\mathbf{d}_2$} ... [k]<s>{#$\mathbf{d}_k$} Only give me the answer and do not output any other words. Question: {#**q**} Answer:
BubbleRAG	Write a high-quality answer for the given question using only the provided search results (some of which might be irrelevant). Only give me the answer and do not output any other words. Question: {#**q**} [1]{#$\mathbf{d}_1$}<b> [2]{#$\mathbf{d}_2$}<b> ... [k]{#$\mathbf{d}_k$}<b> Only give me the answer and do not output any other words. Question: {#**q**} Answer:

Table 9: Prompt templates of different methods. "Comp." means compression-based methods, including LongLLMLingua and LLMLingua-2. "<s>" and "<b>" are the placeholders for semantic embedding and bookmark tokens, respectively.