ARISE: Active Reasoning with Internalized knowledge and Scalable parameter Evolution

Anonymous ACL submission

Abstract

Multi-hop question answering (QA) requires effectively integrating scattered evidence for complex reasoning. While Retrieval-Augmented Generation (RAG) is commonly used, the traditional "retrieve-then-generate" paradigm often loses intermediate reasoning details, and fixed inference parameters hinder real-time adaptation. To address this, we propose **ARISE** (**A**ctive **R**easoning with **I**nternaliz-ed knowledge and **S**calable parameter **E**volution). **ARISE** first internalizes facts into "knowledge index parameters" via masked entity modeling, offering more precise activation and lower overhead than In-Context Learning (ICL). During inference, its Active Reasoning mechanism triggers lightweight parameter refreshing to enable "reasoning-while-updating," resolving conflicts between stale knowledge and new evidence. Additionally, a memory filtering strategy periodically merges high-value Low-Rank Adaptation (LoRA) weights into long-term memory to prevent catastrophic forgetting. Experiments on four datasets confirm that **ARISE** significantly improves reasoning performance and robust logical consistency in multi-hop QA.

1 Introduction

Traditional question-answering (QA) tasks initially focused on single-hop settings (Rajpurkar et al., 2016). With the increasing complexity of decision-making, multi-hop QA has gradually emerged, requiring systems to emulate human-like reasoning to synthesize information across multiple documents (Yang et al., 2018). Large Language Models (LLMs) (Achiam et al., 2023; GLM et al., 2024; Hui et al., 2024) are often employed for such problems but suffer from inherent limitations, including static knowledge (Komeili et al., 2022) and generative hallucination (Welleck et al., 2019; Wallat et al., 2025). Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Gao et al., 2023; Luo et al., 2023) mitigates these issues by incorporating external information. However, the traditional RAG paradigm relies heavily on injecting long-form contexts (Asai et al., 2024; Guu et al., 2020; Lin et al., 2023). This approach incurs substantial computational overhead, suffers from the “Lost in the Middle” problem (Huang et al., 2023; Liu et al., 2024), and constrains the model’s ability to understand logical associations (Li et al., 2024). Consequently, the model remains in a state of “passive reading,” struggling to transform external knowledge into internalized reasoning logic (Xu et al., 2024b; Zhou et al., 2025). Enabling LLMs to achieve genuine logical comprehension and active reasoning is therefore crucial (Press et al., 2023). Such capabilities not only facilitate complex downstream tasks like multi-hop QA (Cheng et al., 2025), but also significantly improve reasoning ability, interpretability, and the overall reliability of RAG systems in real-world scenarios.

Existing RAG methods have explored ways to improve generated answers. Self-RAG (Asai et al., 2024) evaluates retrieval and generation reliability through self-reflection tokens. FLARE (Jiang et al., 2023) monitors generation confidence, triggering re-retrieval when it falls below a threshold. GraphRAG (Edge et al., 2024) incorporates knowledge graphs to structure unstructured text for logical global retrieval. However, these methods assume fixed model knowledge structures prior to reasoning, relying on implicit parametric knowledge while treating external knowledge merely as temporary context. Consequently, they often struggle to grasp the solution logic of complex problems.

Recently, knowledge internalization via Supervised Fine-Tuning (SFT) (Zhang et al., 2024) has been explored, but it incurs heavy computational overhead, making it impractical for per-query document injection. Another branch, Model Editing (Meng et al., 2022b) attempts to internalize knowledge by modifying specific parameters but typically handles only simple facts. When faced with

multi-hop reasoning or long texts, it often leads to catastrophic forgetting. Furthermore, static model parameters during inference cannot continuously assimilate new knowledge or adapt to dynamic scenarios (Meng et al., 2022a; Wang et al., 2025). Conflicts between stale parametric information and retrieved documents often trigger strong prior biases (Jeong et al., 2024; Shi et al., 2025), causing models to favor memorized "parametric knowledge" over RAG-provided "contextual knowledge" (Huang et al., 2025; Sun et al., 2025). In long-range tasks, the isolation between retrieval and generation prevents models from refining their internal logic by absorbing new evidence.

In contrast to direct SFT or generative-based internalization, our design adopts Masked Entity Modeling (MEM) for knowledge internalization. This approach leverages the surrounding context of masked tokens to internalize factual associations into "knowledge index parameters". The "cloze-test" process encourages the model to capture bidirectional relationships between entities, achieving deeper knowledge accumulation with lower computational overhead. Meanwhile, "reasoning-while-updating" has emerged as a strategy to dynamically align internal knowledge with dialogue contexts, enabling models to perform reasoning while selectively updating a subset of parameters to reconcile outdated knowledge with new evidence.

Consequently, we develop **ARISE** (**A**ctive **R**easoning with **I**nternalized knowledge and **S**calable parameter **E**volution), a framework combining offline parameter internalization and online dynamic parameter refreshing. Specifically, to ensure effective knowledge internalization rather than token-level copying, we augment source documents by (1) incorporating synthetic QA pairs derived from the text; and (2) augmenting the documents by rewriting them multiple times to express the same factual content in different forms. Secondly, core entities are selectively masked to encode factual associations into "knowledge index parameters", enhancing the model's ability to capture and retrieve key facts while preserving semantic coherence. During online inference, an Active Reasoning mechanism evaluates reasoning confidence and identifies knowledge gaps. When necessary, the model is lightly refreshed, enabling dynamic reasoning-while-updating. After each dialogue, high-impact "core memory" segments are retained, and Low-Rank Adaptation (LoRA)(Hu et al., 2022) weights are periodically merged into a permanent state. A memory selection strategy based on usage frequency and contribution mitigates catastrophic forgetting and knowledge conflicts, improving multi-hop reasoning consistency. In summary, our key contributions are as follows:

- We propose an offline parameter internalization strategy that enables lightweight fine-tuning and more accurate activation of factual knowledge compared to traditional RAG.
- We introduce an online *reasoning-while-updating* paradigm, integrating filtered core memories through lightweight parameter refreshing to support continuous learning and long-term knowledge retention.
- We conduct extensive experiments on four benchmark datasets, demonstrating that **ARISE** is a lightweight yet effective approach for complex multi-hop QA across diverse scenarios.

2 Related Work

2.1 Retrieval Augmented Generation

Retrieval-Augmented Generation (RAG) is a widely-used technique for addressing common Large Language Model (LLM) limitations, such as static knowledge and hallucinations, by incorporating external knowledge bases (Jiang et al., 2024). It improves knowledge-intensive tasks (e.g., question answering, fact verification) by retrieving relevant documents and using them as contextual evidence in prompts (Lewis et al., 2020), . Despite its effectiveness, standard RAG typically operates under a “retrieve-then-generate” framework (Gao et al., 2023), in which retrieval is performed once at the beginning of inference and the retrieved content is treated as static context throughout the generation process. This design implicitly assumes that the information required to answer a query can be fully captured by a single retrieval step. However, in real-world scenarios, user queries often exhibit dynamic complexity and demand reasoning capabilities that extend beyond surface-level semantic matching. In such cases, the correct answer is rarely contained within a single document; instead, it is distributed across multiple, heterogeneous pieces of evidence that must be iteratively discovered and logically composed.

As a result, conventional “single-hop” RAG methods struggle to model the dependencies and

interactions among multiple retrieved documents, leading to incomplete or inconsistent reasoning outcomes (Huang et al., 2023). These limitations become particularly pronounced in complex question answering settings, where intermediate conclusions are required to guide subsequent information retrieval. This observation highlights the need for more advanced mechanisms that support sequential reasoning, evidence aggregation, and iterative knowledge integration, thereby motivating the development of Multi-hop Question Answering (Multi-hop QA) approaches.

2.2 Multi-hop QA

Multi-hop QA is an extension of conventional question answering, requiring the system to gather information from multiple documents, often in an iterative manner, and to comprehensively contextualize this information in order to answer more complex queries(Press et al., 2023). In recent research, several types of methods have been proposed to address this type of problem. IRCOT (Trivedi et al., 2023) and CRITIC (Gou et al.) , which apply chain-of-thought reasoning by decomposing problems into sub-problems, retrieving documents, and generating logical sequences until the answer is reached. IM-RAG(Yang et al., 2024) enables LLMs to interact with retrieval systems through multiple "self-questioning" iterations. Adaptive-RAG (Jeong et al., 2024) and SIM-RAG(Yang et al., 2025) train classifiers to categorize query complexity and apply dynamic retrieval strategies. FLARE (Jiang et al., 2023) and DRAGIN (Su et al., 2024) trigger retrieval based on model internal states (e.g., low token confidence). However, PRAG (Su et al., 2025) argues that LLM knowledge is primarily stored in its parameters, while context injection only affects key-value pairs in the attention mechanism and cannot be deeply understood and flexibly utilized by the model like internal parameter knowledge. Therefore, it proposes transforming documents into a representation that can be seamlessly integrated into the LLM's parameters, thereby potentially enhancing model comprehension.

3 Methodology

3.1 Problem Formulation and Overview

Given a complex multi-hop query q and an external knowledge base $\mathcal{D}$, the goal is to leverage a LLM f_θ with base parameters θ_{base} to synthesize evidence across scattered documents $D = \{d_1, d_2, \ldots, d_k\} \subseteq \mathcal{D}$ and generate an accurate answer A. A retrieval module $\mathcal{R}$ computes relevance scores $S = \{S_{d_1}, S_{d_2}, \ldots, S_{d_k}\}$ for each document relative to q (Yu and Ananiadou, 2024).

As illustrated in Figure 1, **ARISE** addresses this problem through a three-stage pipeline: (1) **Offline Parameter Internalization via Masked Modeling**: augmenting documents and optimizing lightweight parameters to internalize factual knowledge; (2) **Online Active Reasoning and Dynamic Refreshing**: dynamically updating parameters via real-time uncertainty probing during generation; and (3) **Knowledge Consolidation and Memory Write-back**: consolidating high-value core adapters into long-term persistent memory.

3.2 Offline Parameter Internalization via Masked Modeling

Traditional RAG typically injects retrieved documents directly into the prompt, affecting only attention key-value states while leaving core parameters essentially unchanged (Nanda et al., 2023; Mallen et al., 2023). This limits deep logical synthesis. This module shifts the paradigm by encoding key entities, relations, and multi-hop dependencies directly into the Feed-Forward Network (FFN) parameters (Allen-Zhu and Li) $\Delta\theta_{\text{int}}$.

3.2.1 Knowledge-Aware Document Augmentation

To ensure the model internalizes flexible factual logic rather than verbatim token sequences, we augment each document $d \in D$ via two complementary strategies: (1) **Multi-form Rewriting**: We prompt the model f_θ to rewrite the original document $d \in D$ into multiple versions $({d_{rew}}^1, {d_{rew}}^2, \ldots, {d_{rew}}^n)$ that maintain the same factual content while utilizing diverse linguistic expressions. This process facilitates the decoupling of factual associations from specific syntactic structures. (2) **Synthetic QA Generation**: We prompt the model f_θ to generate a series of synthetic question-answer pairs $\{({q_{syn}}^1, {a_{syn}}^1), ({q_{syn}}^2, {a_{syn}}^2), \ldots, ({q_{syn}}^m, {a_{syn}}^m)\}$ based on the document content, m and n are adjustable hyperparameters. In this context, a_{syn} incorporates detailed reasoning paths, guiding the model to simulate the multi-hop reasoning process during the parameter training phase. The final augmented document D_{aug} combines the augmented results of the above two steps and can

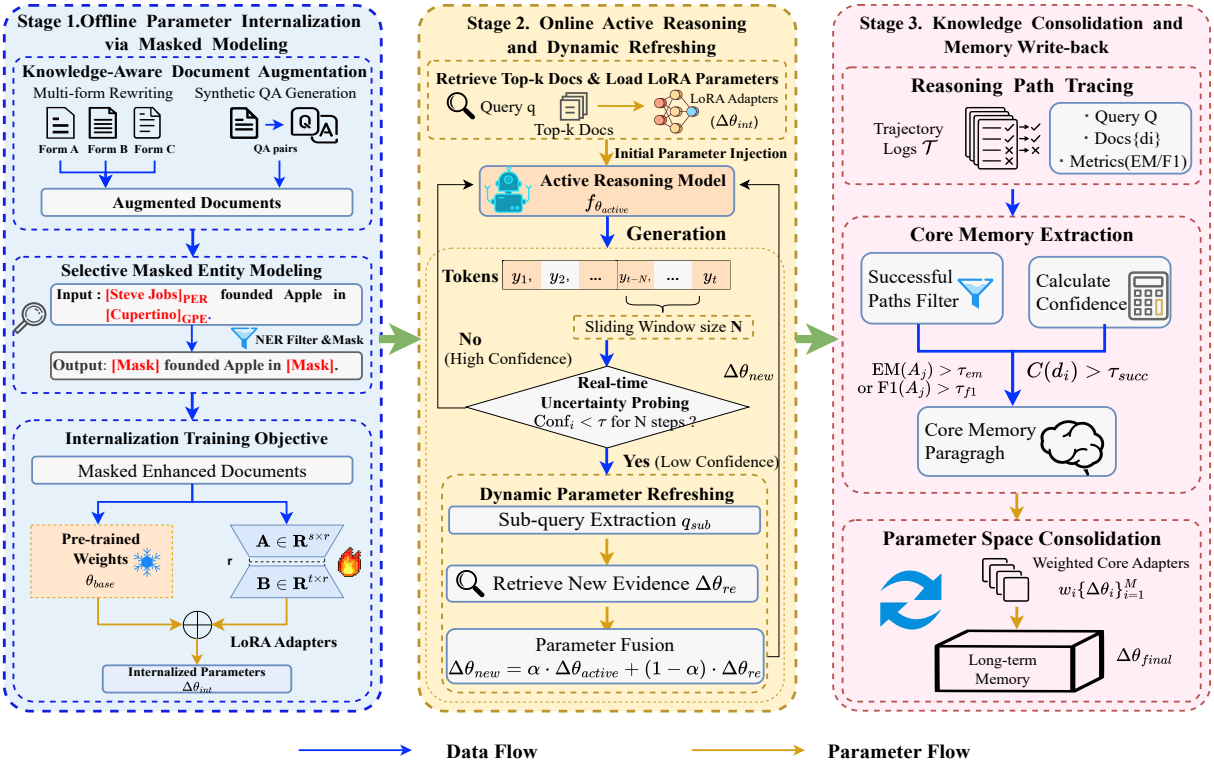

Figure 1: The parameter evolution framework of ARISE. The model evolves through three phases: (1) Offline Parameter Internalization via Masked Modeling: Facts are encoded into internalized parameters ($\Delta\theta_{int}$) via selective masked modeling; (2) Online Active Reasoning and Dynamic Refreshing: During inference, parameters are updated in real-time ($\Delta\theta_{new}$) to absorb retrieved evidence when uncertainty arises; and (3) Knowledge Consolidation and Memory Write-back: High-value core adapters are weighted and merged into long-term memory ($\Delta\theta_{final}$).

be illustrated as:

$$D_{\text{aug}} = \{(d^i_{\text{rew}}, q^j_{\text{syn}}, a^j_{\text{syn}}) \mid i \in \{1,\ldots,n\}, j \in \{1,\ldots,m\}\} \quad (1)$$

3.2.2 Selective Masked Entity Modeling

To enhance the model's perception of core facts, we introduce Masked Entity Modeling (MEM). We utilize a pre-trained Named Entity Recognition (NER) (Ma and Hovy, 2016) model and noun phrase extraction from spaCy (Honnibal et al., 2020) to extract the core informational unit set E for each $d_{\text{aug}} \in D_{\text{aug}}$:

$$E = \{e \mid e \in \mathcal{M}_{\text{ner}}(d_{\text{aug}}) \cup \mathcal{G}_{\text{np}}(d_{\text{aug}}, \text{PROPN})\} \quad (2)$$

where $\mathcal{M}_{\text{ner}}(\cdot)$ extracts named entities (e.g., PERSON, ORG, GPE), and $\mathcal{G}_{\text{np}}(\cdot, \text{PROPN})$ extracts noun phrases containing at least one proper noun.

To balance contextual coherence with knowledge internalization, we randomly sample a subset of entity indices $\mathcal{M}$ governed by a masking ratio ρ:

$$|\mathcal{M}| = \lceil \rho \cdot |E| \rceil \rho \cdot |E| \rceil \quad (3)$$

The selected entity strings in d_{aug} are substituted with a special [MASK] token via a mapping function $\mathcal{F}(\cdot)$ to yield the masked enhanced document d_{mask}:

$$d_{\text{mask}} = \mathcal{F}(d_{\text{aug}}, \{e_h \mid h \in \mathcal{M}\}, \texttt{[MASK]}) \quad (4)$$

This cloze-style objective compels the model to capture bidirectional logical dependencies between entities rather than shallow sequence predictions.

3.2.3 Internalization Training Objective

To efficiently internalize knowledge without compromising the global generality of f_θ, we apply Low-Rank Adaptation (LoRA) to the projection layers of the model's FFN. Given an original weight matrix $W \in \mathbb{R}^{s\times t}$ with hidden dimension s and intermediate dimension t, we introduce low-rank matrices $A \in \mathbb{R}^{s\times r}$ and $B \in \mathbb{R}^{t\times r}$ (where rank $r \ll \min(s,t)$). The modified weight matrix W' is defined as:

$$W' = W + \Delta W = W + AB^\top \quad (5)$$

During training, θ_{base} remains frozen, and the lightweight parameter set $\Delta\theta_{\text{int}} = \{A, B\}$ is opti-

mized by minimizing the negative log-likelihood loss:

$$\mathcal{L}(\Delta\theta_{\text{int}}) = -\sum_{e} \log P(e \mid d_{\text{mask}}; \theta_{\text{base}}, AB^{\top}) \tag{6}$$

where P is the conditional probability of predicting the correct masked entity e. Upon completion of this stage, the explicit masked document transitions from d_{mask} to an internalized state $\Delta\theta_{int}$, rendering the LoRA subspace an efficient container for factual activation.

3.3 Online Active Reasoning and Dynamic Refreshing

Static parameters cannot assimilate real-time evidence or resolve conflicts between stale parametric memory and newly retrieved documents during inference (Du et al., 2024). We introduce a "Reasoning-while-Updating" paradigm to dynamically refresh parameters based on real-time generation uncertainty (Xu et al., 2024a).

3.3.1 Initial Parameter Injection

At the onset of inference for query q, the retrieval module $\mathcal{R}$ identifies the top-k documents $\{d_1, \ldots, d_k\}$. **ARISE** loads their corresponding pre-trained LoRA adapters $\Delta\theta_{\text{int}}$ and initializes the active reasoning parameters θ_{active} by merging them with the base parameters:

$$\theta_{\text{active}} = \theta_{\text{base}} + \text{merge}(\Delta\theta_1, \Delta\theta_2, \ldots, \Delta\theta_k) \tag{7}$$

This establishes a dense contextual activation of query-relevant facts.

3.3.2 Real-time Uncertainty Probing

During generation, the system continuously tracks token-level uncertainty to identify knowledge gaps. For the t-th generated token y_t, given context $(q + y_{<t})$, the token confidence score $Conf_t$ is computed via a softmax operation over the final position logits:

$$Conf_t = \max(\text{Softmax}(y_t \mid y_{<t}, q; \theta_{\text{active}})) \tag{8}$$

A sliding window of length N monitors the confidence sequence. A parameter refresh is triggered if all confidence values within the window fall below a preset threshold τ, defined by the indicator function $\mathbf{1}_{\text{trigger}}$:

$$\mathbf{1}_{\text{trigger}} = \begin{cases} 1, & \text{if } \forall i \in \{t-N+1, \ldots, t\}, \\ & \quad Conf_i < \tau \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

Otherwise, the answer will be generated directly.

3.3.3 Dynamic Parameter Refreshing

When When a trigger event occurs ($\mathbf{1}_{trigger} = 1$), a sub-query q_{sub} is extracted from the current context $y_{<t}$ to represent the missing information required for the next reasoning hop. The system retrieves a new set of relevant parameter fragments $\Delta\theta_{\text{re}}$ from the external base and performs a weighted fusion to update the active reasoning weights:

$$\Delta\theta_{\text{new}} = \alpha \cdot \Delta\theta_{\text{active}} + (1-\alpha) \cdot \Delta\theta_{\text{re}} \tag{10}$$

where $\alpha \in [0, 1]$ balances the existing reasoning state and new evidence. $\Delta\theta_{\text{new}}$ enables the model to smoothly generate tokens with updated evidence.

3.4 Knowledge Consolidation and Memory Write-back

Parameter updates generated by multi-hop inference are often scattered and temporary. If not effectively solidified, this can lead to fragmented inference logic, and frequent parameter switching can result in huge computational overhead (Sheng et al., 2023). This module explicitly prevents catastrophic forgetting and enables continuous self-improvement of the model by selecting "core memories" with relatively high usage frequency and high contribution and periodically merging them into long-term parameters.

3.4.1 Reasoning Path Tracing

We record structural logs of every reasoning trajectory $\mathcal{T}$ during inference, formally defined as a tuple:

$$\mathcal{T} = \{q, \{d_i\}, A, \textit{Metrics}\} \tag{11}$$

where $\{d_i\}$ (or their adapters $\Delta\theta_i$) are the knowledge segments utilized in the reasoning chain, A is the final answer, and *Metrics* denotes evaluation scores such as Exact Match (EM) or F1.The recorder also logs timestamps and model output confidence, providing a granular trace of how specific knowledge segments contribute to successful reasoning steps.

3.4.2 Core Memory Extraction

The system first determines trajectory validity against ground-truth metrics using a success indicator function $I(\mathcal{T}_j)$:

$$I(\mathcal{T}_j) = \begin{cases} 1, & \text{if EM}(A_j) > \tau_{\text{em}} \text{ or F1}(A_j) > \tau_{\text{f1}} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

For each unique segment d_i, we aggregate two statistics across logs: total usage count $C_{\text{total}}(d_i)$,

and successful activation count $C_{\text{succ}}(d_i)$ (where $I(\mathcal{T}_j) = 1$). The empirical success rate $C(d_i)$ is calculated as:

$$C(d_i) = \frac{C_{\text{succ}}(d_i)}{C_{\text{total}}(d_i)} \tag{13}$$

If $C(d_i)$ exceeds a threshold τ_{succ}, the segment is marked as a core memory paragraph.

3.4.3 Parameter Space Consolidation

Given M extracted core knowledge segments and their adapters $\{\Delta\theta_i\}_{i=1}^{M}$, we compute an optimal fusion weight w_i using a softmax function with a temperature parameter β:

$$w_i = \frac{\exp\left(\frac{\log(1+C_{\text{total}}(d_i))\cdot C(d_i)}{\beta}\right)}{\sum_{j=1}^{M}\exp\left(\frac{\log(1+C_{\text{total}}(d_j))\cdot C(d_j)}{\beta}\right)} \tag{14}$$

where the logarithmic scaling of $C_{\text{total}}(d_i)$ prevents high-frequency segments from over-dominating, and $C(d_i)$ prioritizes precise reasoning contributions. The final persistent consolidated parameter update $\Delta\theta_{\text{final}}$ is derived via:

$$\Delta\theta_{\text{final}} = \sum_{i=1}^{M} w_i \cdot (A_i B_i^{\top}) \tag{15}$$

Because all adapters share the same low-rank subspace r, this consolidation effectively writes back transient online learning into persistent internal knowledge.

4 Experiments

In this section, we first detail the experimental setup and implementation, and then report the overall performance and in-depth analysis of our proposed method.

4.1 Experimental Setup

4.1.1 Datasets and Evaluation Metric

We evaluate **ARISE** on four complex multi-hop QA benchmarks: **2WikiMultihopQA (2WQA)**, **HotpotQA**, **PopQA**, and **Complex WebQuestions (CWQ)**. These datasets vary in reasoning depth and domain diversity. Following standard protocols, we sample 300 questions from each subset and use the F1 score as our primary evaluation metric. Detailed dataset descriptions are provided in Appendix A.1.

4.1.2 Implementation Details and Baselines

Our framework is implemented across Qwen2.5-1.5B, Llama-3.2-1B, and Llama-3-8B. In the offline stage, we set the masking ratio $\rho = 0.7$ and LoRA rank $r = 2$. For online inference, the retrieval size is $k = 3$, the uncertainty window is $N = 5$ with $\tau = 0.72$, and the fusion coefficient $\alpha = 0.35$. Full implementation and training details can be found in Appendix A.2. To conduct a comprehensive comparison, we evaluate **ARISE** against several prominent retrieval-augmented generation paradigms: **Standard RAG**, **DRAGIN** (Su et al., 2024), and **PRAG** (Su et al., 2025). Furthermore, following (Su et al., 2025), we include two hybrid variants, **PRAG-Combine** and our proposed **ARISE-Combine**, to investigate the synergy between traditional context injection and parametric knowledge. Detailed architectural descriptions and mechanisms for each baseline are provided in Appendix A.3.

4.2 Experimental Results

The experimental results under the three different models are shown in Table 1, where all methods are evaluated using the same prompt template to ensure a fair and controlled comparison, from which we observe that: Firstly, the proposed **ARISE** and **ARISE-Combine** consistently achieve the state-of-the-art (SOTA) performance across the majority of datasets and base models. This demonstrates that actively internalizing and dynamically refreshing knowledge at the parameter level is more effective than purely retrieval-based or static parameterized RAG variants. Specifically, on Llama3.2-1B and Llama3-8B, **ARISE-Combine** secures the best results in terms of "Total" scores for 2WQA and HotpotQA, as well as for the PopQA and CWQ benchmarks. This demonstrates the robustness of our approach across varying model architectures and parameter scales. Secondly, "combine" strategies (both **PRAG-Combine** and **ARISE-combine**) significantly outperform their respective base versions. This indicates that jointly leveraging contextual knowledge and parametric knowledge leads to significant performance improvements, and that our method is fully compatible with traditional context-based approaches without causing conflicts. In particular, **ARISE-Combine** shows clear advantages on PopQA and CWQ, which require robust cross-document reasoning and knowledge composition. Thirdly, as the base model scales from lightweight

Base LLM	Method	2WQA					HotpotQA			PopQA	CWQ
		Compare	Bridge	Inference	Compose	Total	Bridge	Compare	Total		
LLaMA3.2-1B	**Standard RAG**	0.4298	0.3032	0.2263	0.1064	0.2520	0.2110	0.4083	0.2671	0.1839	0.3726
	DRAGIN	0.4556	0.3357	0.1919	0.0901	0.2692	0.1431	0.4015	0.1830	0.1056	0.3900
	PRAG	0.4920	0.3994	0.2185	0.1334	0.2764	0.1602	**0.4493**	0.1999	0.2205	0.3482
	PRAG-Combine	0.5046	**0.4595**	0.2399	0.1357	0.3237	0.2282	0.4217	0.2689	0.2961	0.4101
	ARISE (ours)	0.4952	0.341	0.2495	0.1242	0.3032	0.2171	0.4162	0.2366	0.2339	0.3731
	ARISE-Combine (ours)	**0.5194**	0.4232	**0.2692**	**0.1408**	**0.3421**	**0.2547**	0.4072	**0.2755**	**0.3163**	**0.4286**
Qwen2.5-1.5B	**Standard RAG**	0.3875	0.3884	0.1187	0.0568	0.2431	0.1619	0.3713	0.2073	0.0999	0.2823
	DRAGIN	0.2771	0.1826	0.1025	0.0680	0.1538	0.0801	0.1851	0.0973	0.0548	0.1788
	PRAG	0.4529	0.4494	**0.2072**	0.1372	0.3025	0.1720	0.4623	0.2165	0.1885	0.3280
	PRAG-Combine	0.4053	0.4420	0.1705	0.1154	0.2627	0.2383	0.5037	0.2942	0.2261	0.3495
	ARISE (ours)	**0.4698**	**0.4627**	0.1976	0.1453	**0.3225**	0.1981	0.4639	0.2769	0.2103	0.3139
	ARISE-Combine (ours)	0.4379	0.4536	0.1763	**0.1462**	0.2879	**0.2541**	**0.5219**	**0.3103**	**0.2319**	**0.3537**
LLaMA3-8B	**Standard RAG**	0.5843	0.4794	0.1833	0.0991	0.3372	0.1823	0.3493	0.2277	0.1613	0.3545
	DRAGIN	0.5185	0.4480	0.2664	0.1833	0.3544	0.2618	0.6116	0.2924	0.1772	0.3101
	PRAG	0.6353	0.5437	0.2471	0.1992	0.3932	0.3115	0.6557	0.3563	0.2413	0.4541
	PRAG-Combine	0.6432	0.5556	**0.3160**	0.2339	0.4258	0.4025	**0.6918**	0.4559	**0.3059**	0.4728
	ARISE (ours)	0.6241	0.5641	0.2899	0.2097	0.4196	0.3516	0.6252	0.4184	0.2596	0.4531
	ARISE-Combine (ours)	**0.6720**	**0.5722**	0.3085	**0.2649**	**0.4627**	**0.4219**	0.6389	**0.4925**	0.2847	**0.4939**

Table 1: Main experimental results of **ARISE** and other baseline methods. The best results are in **bold** and the second best results are underlined. As data types are non-uniformly distributed in 2WQA and HotpotQA, the "Total" column reflects the holistic performance of all methods on the complete dataset. Results are averaged over three random runs.

(Llama3.2-1B, Qwen2.5-1.5B) to larger (Llama3-8B), the performance gains of **ARISE** become more pronounced, suggesting that the proposed approach benefits from increased model capacity, while maintaining stable improvements across different reasoning types such as bridge, compare, and inference questions.

4.3 Ablation Study

In this section, we conduct a series of ablation experiments to verify the importance of each module in our methods, as shown in Table 2. **ARISE** consists of three stages, and we first analyze the role of each stage individually. In particular, the data preprocessing in the first stage includes data augmentation (multi-form rewriting and synthetic QA generation) and masked entity modeling. Finally, we separately investigate the impact of these two operations.

4.3.1 The Impact of Data Preprocessing

In the first stage, we performed two data preprocessing steps before transforming external knowledge into parameters that can be injected into the model. First, we use the LLM with task-specific prompts to generate semantically identical rewrites and QA pairs from the original documents, thereby achieving data augmentation. Then, we apply masked entity modeling based on the generated augmented documents. In the ablation experiment, we remove the above preprocessing steps and directly convert the raw documents into parameters that are injected

Base LLM	Method	2WQA	HotpotQA	PopQA	CWQ
LLaMA-1B	**w/o DP**	0.2839	0.2051	0.2112	0.3548
	w/o UP	0.2871	0.2219	0.2205	0.3599
	w/o MW	0.2980	0.2211	0.2231	0.3612
	ARISE	**0.3032**	**0.2366**	**0.2339**	**0.3731**
Qwen-1.5B	**w/o DP**	0.3008	0.2596	0.1989	0.2986
	w/o UP	0.3128	0.2653	0.1993	0.2974
	w/o MW	0.3216	0.2742	0.2091	0.3108
	ARISE	**0.3225**	**0.2769**	**0.2103**	**0.3139**
LLaMA-8B	**w/o DP**	0.4067	0.3957	0.2358	0.4418
	w/o UP	0.3982	0.3943	0.2453	0.4385
	w/o MW	0.4188	0.4093	0.2486	0.4507
	ARISE	**0.4196**	**0.4184**	**0.2596**	**0.4531**

Table 2: Ablation study of ARISE components across various benchmarks. All metrics reported are F1 scores. “w/o” indicates the corresponding module in ARISE is removed.

into the LLM. As shown in the "w/o DP" row of Table 2, the results indicate that data preprocessing (DP) is crucial to the overall performance of **ARISE**. Among all ablation variants, removing DP (i.e., "w/o DP") generally leads to the most significant performance degradation, particularly on the 2WQA and CWQ benchmarks. This indicates that the DP step uncovers some deeper information and semantic relationships in the original document, which more effectively enhances the model's ability to understand and extract knowledge than using the original document directly. To further decouple the synergistic effects of Data Augmentation (DA) and Masked Entity Modeling (MEM) within the preprocessing stage, we provide a fine-grained

ablation study in Appendix B.1.

4.3.2 The Impact of Uncertainty Probing

In the second stage, we first perform an initial injection of the $top - k$ document parameters retrieved based on the query. During generation, we then continuously monitor token-level uncertainty to determine when the internal knowledge is insufficient to support the current reasoning step, thereby triggering updates with new knowledge required for reasoning. In the ablation experiment, we remove the approach of updating while refreshing and instead use a method of injecting all the $top - k$ document parameters at once for inference. As shown in the "w/o UP" row of Table 2, removing the dynamic uncertainty probing leads to a performance decline across all three base models. For instance, on the LLaMA-1B model, the F1 score for the 2WQA task drops from 0.3032 to 0.2871, and for the CWQ task, it decreases from 0.3731 to 0.3599. This indicates the necessity of dynamically supplementing knowledge as inference progresses in multi-hop tasks. Compared to simply injecting all $top - k$ document parameters at once, the "Reasoning-while-Updating" paradigm is more effective. When the model's internal knowledge becomes insufficient, dynamically triggering the refresh of new knowledge effectively ensures the logical consistency of the inference chain and the real-time nature of the evidence.

4.3.3 The Impact of Memory Write-back

In the third stage, reasoning paths and scores are recorded after each QA session, and then the "core memories" characterized by high utilization frequency and significant contribution are selected and periodically merged into long-term parameters. In the ablation experiment, we remove the parameter merging of core memories after the session ends, and directly destroy the LoRA parameters generated during the reasoning process. As shown in the "w/o MW" row of Table 2, although removing memory write-back leads to a relatively small performance drop in practice compared to other modules, its purpose is not to boost immediate accuracy but to serve as a post-processing step in the reasoning stage to improve computational efficiency. During multi-hop reasoning, online parameter updates are often transient and highly dispersed. Without effective consolidation, this behavior can eventually result in fragmented reasoning logic. Under the "w/o MW" setting, the model must repeatedly trigger parameter refreshes for each inference, with each refresh introducing an average overhead of about 0.37 seconds. For complex tasks that require multiple retrieval and update cycles, this latency accumulates significantly. By writing back these scattered online learning outcomes and integrating them into persistent internal knowledge, the long-term performance of the model can be improved, thereby enhancing its continuous learning ability.

4.4 Hyperparameter Sensitivity Analysis

We conduct sensitivity analyses on three critical hyperparameters using the LLaMA-3.2-1B model: the masked entity ratio ρ, the confidence threshold τ, and the fusion coefficient α. Our findings indicate that moderate masking ($\rho \in [0.6, 0.8]$) balances contextual coherence with memory strength ; a mid-range threshold ($\tau \approx 0.7$) avoids excessive parameter churn while rectifying knowledge gaps ; and an intermediate fusion weight ($\alpha \approx 0.3$–0.4) ensures steady logical continuity. The detailed discussion and corresponding curves (Figure 2) are detailed in Appendix C.

5 Conclusion

In this paper, we propose **ARISE**, a knowledge enhancement model that explicitly addresses knowledge gaps and static parameter issues in complex multi-hop QA. Unlike traditional RAG methods that rely on passive context injection, **ARISE** achieves active reasoning through a three-stage evolutionary framework, including Offline Parameter Internalization, Online Active Reasoning and Core Memory Write-back. Experimental results demonstrate that **ARISE** achieves state-of-the-art (SOTA) performance on the majority of the four mainstream multi-hop question-answering benchmark datasets, proving its robustness across language models of varying sizes (1B to 8B). Furthermore, **ARISE's** excellent compatibility with traditional retrieval methods (**ARISE-Combine**) clearly showcases its significant potential in handling complex real-world scenarios. Future work may explore more fine-grained uncertainty estimation strategies, scalable adapter management for large and heterogeneous knowledge bases, and tighter integration between retrieval, reasoning, and parameter evolution.

Limitations

Although **ARISE** achieves strong performance in multi-hop question answering, it has several limitations. First, the framework incurs inference latency and computational overhead, as each dynamic parameter refresh introduces an average overhead of about 0.37 seconds , which accumulates significantly over multiple reasoning cycles. Second, managing a massive number of dispersed, lightweight LoRA adapters poses a scalability challenge when extending the system to large and heterogeneous external knowledge bases. Finally, the sliding-window uncertainty probing remains relatively coarse-grained , and the framework's success heavily relies on high-quality data preprocessing, meaning any noise or sub-optimal outputs from the LLM rewriting or entity masking stages can undermine downstream reasoning stability.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction, 2024. *URL https://arxiv. org/abs/2309.14316*.

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

Rong Cheng, Jinyi Liu, Yan Zheng, Fei Ni, Jiazhen Du, Hangyu Mao, Fuzheng Zhang, Bo Wang, and Jianye Hao. 2025. Dualrag: A dual-process approach to integrate reasoning and retrieval for multi-hop question answering. *arXiv preprint arXiv:2504.18243*.

Mingzhe Du, Anh Tuan Luu, Bin Ji, and See-Kiong Ng. 2024. From static to dynamic: knowledge metabolism for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23784–23786.

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

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, and 1 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. Critic: Large language models can self-correct with tool-interactive critiquing, 2024. *URL https://arxiv. org/abs/2305.11738*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, and 1 others. 2020. spacy: Industrial-strength natural language processing in python.

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

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv:2403.14403*.

Xuhui Jiang, Yuxing Tian, Fengrui Hua, Chengjin Xu, Yuanzhuo Wang, and Jian Guo. 2024. A survey on large language model hallucination via a creativity perspective. *arXiv preprint arXiv:2402.06647*.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval

augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In *Proceedings of the 60th annual meeting of the Association for Computational Linguistics (Volume 1: Long papers)*, pages 8460–8478.

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

Xiang Li, Shizhu He, Fangyu Lei, JunYang JunYang, Tianhuang Su, Kang Liu, and Jun Zhao. 2024. Teaching small language models to reason for knowledge-intensive multi-hop question answering. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7804–7816.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, and 1 others. 2023. Ra-dit: Retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023. Dr. icl: Demonstration-retrieved in-context learning. *arXiv preprint arXiv:2305.14128*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

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.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Meta. 2024a. Llama-3.2-1b-instruct. Accessed: 2024-09.

Meta. 2024b. Meta-llama-3-8b-instruct. Accessed: 2024-04.

Neel Nanda, Senthooran Rajamanoharan, Janos Kramar, and Rohin Shah. 2023. Fact finding: Attempting to reverse-engineer factual recall on the neuron level. In *Alignment Forum*, page 6.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc.

Ying Sheng, Shiyi Cao, Dacheng Li, Coleman Hooper, Nicholas Lee, Shuo Yang, Christopher Chou, Banghua Zhu, Lianmin Zheng, Kurt Keutzer, and 1 others. 2023. S-lora: Serving thousands of concurrent lora adapters. *arXiv preprint arXiv:2311.03285*.

Teng Shi, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Han Li. 2025. Retrieval augmented generation with collaborative filtering for personalized text generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1294–1304.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. Dragin: dynamic retrieval augmented generation based on the information needs of large language models. *arXiv preprint arXiv:2403.10081*.

Weihang Su, Yichen Tang, Qingyao Ai, Junxi Yan, Changyue Wang, Hongning Wang, Ziyi Ye, Yujia Zhou, and Yiqun Liu. 2025. Parametric retrieval augmented generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1240–1250.

Zhongxiang Sun, Qipeng Wang, Weijie Yu, Xiaoxue Zang, Kai Zheng, Jun Xu, Xiao Zhang, Yang Song, and Han Li. 2025. Rearter: Retrieval-augmented reasoning with trustworthy process rewarding. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1251–1261.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. *arXiv preprint arXiv:1803.06643*.

Qwen Team and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3).

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 10014–10037.

Jonas Wallat, Maria Heuss, Maarten de Rijke, and Avishek Anand. 2025. Correctness is not faithfulness in retrieval augmented generation attributions. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, pages 22–32.

Yuhao Wang, Ruiyang Ren, Yucheng Wang, Wayne Xin Zhao, Jing Liu, Hua Wu, and Haifeng Wang. 2025. Unveiling knowledge utilization mechanisms in llm-based retrieval-augmented generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1262–1271.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024a. Knowledge conflicts for llms: A survey. *arXiv preprint arXiv:2403.08319*.

Zhipeng Xu, Zhenghao Liu, Yibin Liu, Chenyan Xiong, Yukun Yan, Shuo Wang, Shi Yu, Zhiyuan Liu, and Ge Yu. 2024b. Activerag: Revealing the treasures of knowledge via active learning. *CoRR*.

Diji Yang, Jinmeng Rao, Kezhen Chen, Xiaoyuan Guo, Yawen Zhang, Jie Yang, and Yi Zhang. 2024. Im-rag: Multi-round retrieval-augmented generation through learning inner monologues. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 730–740.

Diji Yang, Linda Zeng, Jinmeng Rao, and Yi Zhang. 2025. Knowing you don't know: Learning when to continue search in multi-round rag through self-practicing. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1305–1315.

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.

Zeping Yu and Sophia Ananiadou. 2024. Neuron-level knowledge attribution in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3267–3280.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*.

Zhanke Zhou, Xiao Feng, Zhaocheng Zhu, Jiangchao Yao, Sanmi Koyejo, and Bo Han. 2025. From passive to active reasoning: Can large language models ask the right questions under incomplete information? *arXiv preprint arXiv:2506.08295*.

A Experimental Setup

A.1 Detailed dataset descriptions

ARISE was evaluated on four complex multi-hop QA datasets, including **2WikiMultihopQA (2WQA)** (Ho et al., 2020), **HotpotQA** (Yang et al., 2018), **PopQA** (Mallen et al., 2023), and **ComplexWebQuestions (CWQ)** (Talmor and Berant, 2018). They differ in reasoning depth, document structure, and domain diversity. Specifically, **2WQA** is a large-scale multi-hop question answering dataset that requires reasoning across multiple entity-linked Wikipedia articles, emphasizing cross-document entity relation reasoning. **HotpotQA** is a multi-hop QA benchmark constructed from Wikipedia, requiring joint reasoning over multiple supporting documents with explicitly annotated supporting facts. **PopQA** is a knowledge-intensive question answering dataset derived from popular factual queries, focusing on long-tail knowledge coverage and open-domain factual recall. **CWQ** is a complex question answering dataset generated from real web queries, requiring multi-step reasoning and decomposition over heterogeneous web-based knowledge sources. **2WQA** and **HotpotQA** are further categorized according to reasoning type, with **2WQA** divided into four categories and **HotpotQA** into two. Following (Su et al., 2025), we select the first 300 questions from each dataset subset for evaluation. For standard evaluation protocols, we adopt the F1 score as the evaluation metric to measure the overlap between predicted answers and gold-standard answers.

A.2 Implementation and training details

To evaluate the performance of our method across models with different types and parameter scales,

we conduct experiments on three LLMs: Qwen2.5-1.5B-Instruct (Team et al., 2024), Llama-3.2-1B-Instruct (Meta, 2024a), and Llama-3-8B-Instruct (Meta, 2024b). For all experiments, we utilize BM25 (Robertson and Zaragoza, 2009) with Elasticsearch as the retriever, which has demonstrated strong performance in RAG. The retrieval corpus is constructed from the Wikimedia dump publicly available for each dataset.

In the offline parameter internalization stage, each document is rewritten once and three question–answer pairs are generated. Selective masked entity modeling is applied with a masking ratio $\rho = 0.7$. LoRA adapters are injected into the FFN layers with rank $r = 2$ and trained for one epoch using AdamW (Loshchilov and Hutter, 2017) with a learning rate of $3e - 4$; the scaling factor α is set to 32, while the base model parameters remain frozen. During online active reasoning, the top-k retrieval size is set to $k = 3$, uncertainty is monitored using a sliding window of length $N = 5$ with confidence threshold $\tau = 0.72$, and dynamic parameter refreshing is performed via weighted fusion with $\alpha = 0.35$. For knowledge consolidation, successful trajectories are filtered using thresholds $\tau_{em} = 0.5$ and $\tau_{f1} = 0.6$, and core adapters are merged with a temperature parameter tuned among [0.5,2.0]. All experiments are repeated three times with random seeds, and results are averaged. Experiments are conducted on four NVIDIA RTX 3090 GPUs with 24GB memory each, using FP16 mixed-precision training and inference.

A.3 Baselines

To conduct a comprehensive comparison, we choose the following RAG methods as baselines:

- **Standard RAG** feeds the query and retrieved context into the LLM as a prompt to generate a grounded response.
- **DRAGIN** (Su et al., 2024) dynamically invokes external knowledge during generation when the model encounters semantically important and high-impact tokens, using its internal state and context to form more precise queries.
- **PRAG** (Su et al., 2025) is a new RAG paradigm, the core idea of which is to transform documents into representations that can be seamlessly inserted into LLM parameters.
- **ARISE**. Our proposed method is a knowledge-augmented model based on offline parameter internalization and online dynamic parameter evolution , which merges high-value LoRA weights into long-term memory.

Following (Su et al., 2025), we introduce two additional baselines (**PRAG-Combine** and **ARISE-Combine**) by integrating context injection from traditional RAG with parameter injection to investigate the mutual influence between them.

B Ablation Study

B.1 The Impact of Data Augmentation and Masked Entity Modeling

In this section, we further deconstruct the two core sub-modules of Data Preprocessing: Data Augmentation (DA) and Mask-ed Entity Modeling (MEM), analyzing their specific contributions to the model's knowledge extraction capabilities. We conduct experiments under a total of four settings: (1) removing all modules, (2) removing only the data augmentation module, (3) removing only the masked entity modeling module, and (4) retaining all modules. The results are shown in Table 3.

Firstly, across all settings, retaining both DA and MEM ("with Both") consistently yields the best performance in most cases. When both sub-modules are removed ("w/o Both"), the model exhibits the most significant drop in F1 scores across all benchmarks. For example, on the Llama-1B model, performance on the 2WQA task sharply decreases from 0.3032 to 0.2839. This indicates that the diverse expressions introduced by DA and the the deep logical mining enabled by MEM play an irreplaceable synergistic role in the process of knowledge internalization. Secondly, removing MEM ("w/o MEM") generally leads to a more severe performance degradation than removing only DA ("w/o DA"). This validates that the selective masked entity modeling, by framing the task as a cloze-style objective, forces the model to move beyond simple token-level prediction and instead capture the bidirectional logical dependencies between entities, thereby more accurately identifying the core nodes that carry factual and logical information. Finally, the experimental data suggests that even in the absence of MEM, DA improves the model's adaptability to complex queries by providing a richer simulation of multi-hop reasoning paths during the training phase. In conclusion, these two steps are complementary and both are crucial for

the effective internalization of external knowledge.

Base LLM	Method	2WQA	HotpotQA	PopQA	CWQ
LLaMA-1B	w/o Both	0.2839	0.2051	0.2112	0.3548
	w/o DA	**0.3041**	0.2312	0.2307	0.3716
	w/o MEM	0.2925	0.2263	0.2297	0.3628
	with Both	0.3032	**0.2366**	**0.2339**	**0.3731**
Qwen-1.5B	w/o Both	0.3008	0.2596	0.1989	0.2986
	w/o DA	0.3219	0.2753	**0.2127**	0.3110
	w/o MEM	0.3156	0.2674	0.2059	0.3083
	with Both	**0.3225**	**0.2769**	0.2103	**0.3139**
LLaMA-8B	w/o Both	0.4067	0.3957	0.2358	0.4418
	w/o DA	0.4174	0.4128	0.2541	0.4509
	w/o MEM	0.4128	0.4081	0.2498	0.4493
	with Both	**0.4196**	**0.4184**	**0.2596**	**0.4531**

Table 3: Ablation study of Data Preprocessing modules across various benchmarks. All metrics reported are F1 scores. "w/o" indicates the corresponding module is removed.

C Further Analyses

To further understand the behavior and robustness of **ARISE**, we conduct a series of hyperparameter sensitivity analyses as shown in Figure 2. These experiments aim to examine how key design choices influence knowledge internalization strength and reasoning stability. Unless otherwise specified, all experiments are conducted on LLaMA-3.2-1B , while keeping other hyperparameters fixed as described in Section 4.1.2.

C.1 Effect of the Masked Entity Ratio ρ

In the offline parameter internalization stage, the masking ratio ρ (Eq.3) controls the proportion of core entities replaced by the [MASK] token. This hyperparameter directly determines the balance between forcing factual reconstruction and preserving contextual coherence. We vary ρ in the range $[0.3, 0.9]$ and observe the following trends:

(1) Low masking ratio (ρ = 0.3) results in limited performance gains. With most entities explicitly visible, the model can rely on shallow token copying rather than learning latent factual associations. As a consequence, the internalized parameters $\Delta\theta_{int}$ exhibit weak generalization and provide limited benefits during multi-hop reasoning. (2) Excessive masking (ρ = 0.9) leads to noticeable performance degradation. Over-masking destroys local syntactic and semantic cues, making the reconstruction task overly ambiguous. This not only increases training difficulty but also introduces noise into the learned LoRA subspace, weakening downstream reasoning stability. (3) Moderate masking ratios (ρ = [0.6,0.8]) consistently yield the best performance across datasets. In this regime, the model is sufficiently challenged to infer masked entities using relational context, encouraging the formation of robust "knowledge index parameters" without severely disrupting sentence semantics.

C.2 Effect of the Reasoning Confidence Threshold τ

In the online active reasoning stage, the confidence threshold τ (Eq.9) determines when dynamic parameter refreshing is triggered. This hyperparameter governs the reactivity of the "reasoning-while-updating" mechanism. We experiment with the range $[0.3, 0.9]$ and make several observations:

(1) Lower thresholds ($\tau \leq 0.5$) make refresh events rare. In this case, the model tends to rely excessively on stale internal knowledge, failing to rectify incorrect intermediate assumptions during multi-hop reasoning. (2) Higher thresholds ($\tau \geq 0.8$) cause frequent refresh triggers. While this increases the model's exposure to new

evidence, it also introduces excessive parameter churn, leading to unstable reasoning trajectories and unnecessary computational overhead. (3) Intermediate values ($\tau \approx 0.7$) achieve the best balance. The model selectively refreshes parameters only when sustained low-confidence signals indicate genuine knowledge insufficiency, resulting in improved reasoning consistency and efficiency.

C.3 Effect of the Fusion Coefficient α

The fusion coefficient α (Eq.10) primarily determines the relative contribution of previously accumulated reasoning state and newly retrieved evidence. Our experiments indicate that:

(1) Small α (e.g., 0.1) overly biases the model toward newly retrieved knowledge, increasing the risk of losing previously established reasoning context. (2) Large α (e.g., 0.7) overly preserves prior parameters, reducing the corrective effect of new evidence. (3) The mid-range ($\alpha \approx 0.3 - 0.4$) consistently performs best, enabling smooth integration of new information while maintaining logical continuity. This behavior highlights the importance of parameter evolution rather than abrupt overwriting during inference.

D Detailed Efficiency and Inference Latency Analysis

To comprehensively address concerns regarding the computational overhead and structural latency

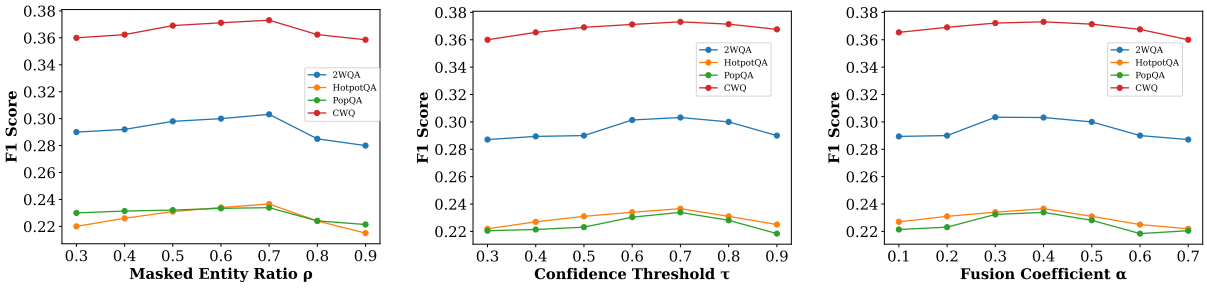

Figure 2: Hyperparameter sensitivity analysis of ARISE. (a) Masked entity ratio ρ. (b) Confidence threshold τ. (c) Fusion coefficient α. All experiments are conducted on LLaMA-3.2-1B.

of ARISE, this section provides a rigorous, fine-grained efficiency analysis. Specifically, we deconstruct the runtime latency, evaluate the wall-clock impact of KV Cache invalidation during dynamic parameter refreshing, compare end-to-end inference costs against competitive baselines, and report the offline training and storage footprints. All benchmarks were evaluated on 4× NVIDIA RTX 3090 GPUs (24GB) using FP16 mixed-precision.

D.1 Granular Latency Breakdown of Parameter Refreshing

As discussed in Section 4.3.3 and Limitations, each dynamic parameter refresh event introduces an average wall-clock overhead of approximately 0.37 seconds. In Table 4, we break down this overhead into three independent sub-steps:

- **Uncertainty Probing & Fragment Retrieval ($\approx$ 0.02s):** The sliding window monitoring ($N = 5$) over token confidence and the lexical retrieval of target parameter fragments $\Delta\theta_{re}$ from the local base.
- **Parameter Fusion ($\approx$ 0.05s):** The element-wise weighted addition to update the active reasoning weights via $\Delta\theta_{new} = \alpha\Delta\theta_{active} + (1-\alpha)\Delta\theta_{re}$.
- **Context Re-prefilling ($\approx$ 0.30s):** As rightly pointed out by specialized literature, modifying model weights mid-generation fundamentally shifts the internal representation space, rendering the existing Key-Value (KV) Cache invalid. Consequently, the system must re-encode the historical context tokens ($q + y_{<t}$) under the new parameter space $\Delta\theta_{new}$ to resume generation. This context re-prefilling constitutes the primary bottleneck ($\approx$ 81.1%) of the refresh overhead.

Component Phase	Latency (s)	Ratio (%)
Uncertainty Probing & Retrieval	0.021	5.7%
Weight Fusion ($\Delta\theta_{new}$)	0.048	13.0%
Context Re-prefilling (KV Re-build)	0.301	81.3%
Total Single Refresh Overhead	**0.370**	**100.0%**

Table 4: Granular latency breakdown of a single dynamic parameter refresh step evaluated on LLaMA-3.2-1B with a context length of 512 tokens.

D.2 End-to-End Inference Efficiency Comparison

To demonstrate that this overhead is sustainable in multi-hop scenarios, we compare the average end-to-end wall-clock execution time per query across four datasets in Table 5.

While ARISE experiences higher latency than Standard RAG due to weight switching and prefix re-computation, it maintains a highly competitive profile compared to DRAGIN. This efficiency stems from our sliding window filtering ($\tau = 0.72, N = 5$), which ensures that parameter refreshing is triggered strictly when sustained low-confidence signals occur. On average, the refresh mechanism is activated only 1.2 to 1.8 times per multi-hop query, effectively preventing excessive parameter churn. Furthermore, after our Stage 3 Memory Write-back consolidates these transient parameters into long-term persistent memory ($\Delta\theta_{final}$), subsequent inferences on similar domains bypass these refresh triggers entirely, dropping the average latency back close to the parametric baseline.

D.3 Offline Preprocessing and Storage Footprint

ARISE restricts the low-rank subspace to an ultra-lightweight setting (rank $r = 2$, training for only 1 epoch), the computational cost per document is negligible.

On a single RTX 3090 GPU, internalizing a

Method	Avg. Refreshes	2WQA (s)	HotpotQA (s)	PopQA (s)	CWQ (s)
LLaMA-3.2-1B					
Standard RAG	0.0	1.04	1.12	0.95	1.21
DRAGIN	2.4 (Retrievals)	1.82	1.95	1.64	2.10
ARISE (Ours)	1.4 (Refreshes)	1.56	1.64	1.47	1.73
LLaMA-3-8B					
Standard RAG	0.0	2.15	2.31	1.98	2.44
DRAGIN	2.6 (Retrievals)	3.42	3.66	3.12	3.89
ARISE (Ours)	1.6 (Refreshes)	2.74	2.90	2.57	3.03

Table 5: End-to-end wall-clock execution time (seconds per query) and average trigger counts across different model scales.

standard Wikipedia document chunk takes approximately 0.12 seconds. Since this preprocessing pipeline is completely decoupled per document, it can be seamlessly parallelized across distributed clusters. Crucially, each resulting LoRA adapter occupies only ≈ 1.2 MB of storage. Utilizing high-throughput adapter serving frameworks like S-LoRA, thousands of distinct knowledge index parameters can be indexed and concurrent loaded into GPU memory with sub-millisecond switching overhead, rendering the framework scalable to heterogeneous, large-scale external knowledge bases.