SR-RAG: Binding Selective Retrieval with Knowledge Verbalization

Di Wu* **Jia-Chen Gu*** **Kai-Wei Chang** **Nanyun Peng**
University of California, Los Angeles
{diwu,kwchang,violetpeng}@cs.ucla.edu, gujc@ucla.edu

Abstract

Selective retrieval improves retrieval-augmented generation (RAG) by reducing distractions from low-quality retrievals and improving efficiency. However, existing methods under-utilize the inherent knowledge of large language models (LLMs), leading to inaccurate retrieval decisions and suboptimal generation performance. To bridge this gap, we propose Self-Routing RAG (SR-RAG), a novel framework that binds selective retrieval with knowledge verbalization. SR-RAG enables an LLM to dynamically self-route between external retrieval and verbalizing its own parametric knowledge. To achieve this, SR-RAG performs multi-task alignment to jointly optimize knowledge source selection, knowledge verbalization, and response generation. Additionally, we introduce a dynamic policy for knowledge source inference via nearest neighbor search to improve the accuracy of knowledge source decision under domain shifts. Fine-tuning three LLMs with SR-RAG significantly improves both their response accuracy and inference latency, reducing retrievals by 29% compared to the strongest selective retrieval baseline while improving performance by 4.9%.

1 Introduction

Retrieval-augmented generation (RAG) enhances large language models (LLMs) with relevant knowledge from external datastores at inference time, improving the performance on tasks requiring up-to-date or domain-specific knowledge (Khandelwal et al., 2020; Lewis et al., 2020; Borgeaud et al., 2022; Ram et al., 2023; Shi et al., 2024). Recently, selective retrieval—an inference strategy that avoids unnecessary retrieval augmentations—has shown promising results in reducing distractions from low-quality retrievals and improving the efficiency of RAG (He et al., 2021; Mallen et al., 2023; Xu et al., 2024; Wu et al., 2024).

However, existing selective retrieval studies overlook a fundamental question:

Are the potentials of the LLM as a knowledge source fully honored in selective retrieval?

When skipping retrieval, current selective retrieval works use a standard yet simplistic fallback: letting the LLM directly generate the response (Mallen et al., 2023; Jeong et al., 2024; Asai et al., 2024; Wu et al., 2024). However, this design limits LLMs' ability to articulate their parametric knowledge before generating a response. We argue that this ability of *knowledge verbalization*, while seeming subtle, critically impacts the success of selective retrieval. First, knowledge verbalization expands the performance upper bound when retrieval is abstained. It has been shown that a knowledgeable LLM is able to directly generate high quality knowledge (Yu et al., 2023) as well as intermediate reasoning paths (Wei et al., 2022; Allen-Zhu and Li, 2023) to benefit the system's performance. This is especially valuable under complex queries, where even compute-intensive retrieval methods can only return noise-prone results. Second, knowledge verbalization enables more accurate selective retrieval decisions. Existing works train retrieval policies by comparing RAG with LLM direct response (Wang et al., 2023; Wu et al., 2024) or resorting to likelihood preferences (He et al., 2021; Xu et al., 2024). By contrast, through explicit knowledge elicitation, knowledge verbalization helps characterize the LLM's capabilities more precisely. Therefore, to realize selective retrieval's full potential, it is imperative to embrace knowledge verbalization.

We propose Self-Routing RAG (SR-RAG), a selective retrieval framework that tightly integrates knowledge verbalization. By reformulating selective retrieval as a *knowledge source selection*

*Equal Contribution

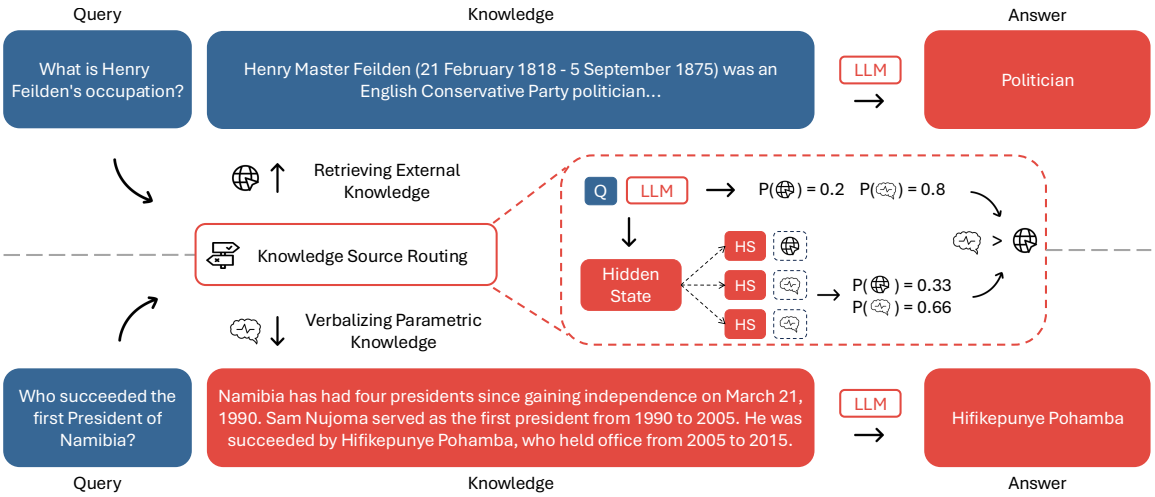

Figure 1: An overview of SR-RAG. Given a user query, the system first selects the most appropriate knowledge source by combining special token prediction with nearest neighbor search. Then, the knowledge is either retrieved from external an external source or self-verbalized by the LLM. Finally, the LLM forms the response based on the query and the knowledge. All the steps are streamlined as a single left-to-right generation pass.

problem, SR-RAG enables an LLM to dynamically self-route between retrieving external knowledge and verbalizing its own parametric knowledge (Figure 1). Observing the limitations of existing frameworks that fine-tune the LLM itself to perform selective retrieval via special tokens (Asai et al., 2024; Wu et al., 2024), we introduce three novel designs. First, SR-RAG performs diverse knowledge verbalization to create more accurate training labels for knowledge source selection (§3.3). Second, SR-RAG incorporates a multi-task alignment approach to jointly optimize the LLM for knowledge source selection, knowledge verbalization, and response generation. Notably, we leverage the diverse verbalized knowledge contexts to perform self-supervised preference alignment for verbalizing high-quality knowledge (§3.3). Finally, existing approaches suffer from poor source decision accuracy at inference stage due to domain shifts and model ability shifts caused by fine-tuning. To bridge this gap, SR-RAG proposes dynamic knowledge source inference via nearest neighbor search, augmenting likelihood-based retrieval decisions with neighboring policy examples in the hidden representation space of the fine-tuned LLM (§3.4). Crucially, SR-RAG's inference remains efficient, requiring only a single left-to-right generation pass.

Using the SR-RAG recipe, we fine-tune Llama-2-7B-Chat (Touvron et al., 2023), Phi-3.5-mini-instruct (Abdin et al., 2024), and Qwen2.5-7B-Instruct (Yang et al., 2024) on a mixture of knowledge-intensive datasets to develop their knowledge source selection ability and enhance the knowledge verbalization quality. Extensive experiments on four knowledge-intensive question answering tasks demonstrate that SR-RAG greatly outperforms both always retrieving and the vanilla selective retrieval approach. Compared to the latter, SR-RAG achieves 7.9%/2.1%/4.7% higher overall performance while performing 26%/40%/21% fewer retrievals across these three LLMs respectively (§5.2). Our analyses further demonstrate that SR-RAG improves both the accuracy of selective RAG decisions (§5.3) and the system's inference efficiency (§5.4). Finally, we carefully ablate the three novel designs and illustrate that all of them are crucial to the strong performance SR-RAG (§5.5).

2 Related Work

Selective Retrieval To enhance the efficiency of RAG systems and avoid potentially harmful retrievals, several works have proposed to selectively skip retrieval augmentation, which we call selective retrieval following Xu et al. (2024) and Wu et al. (2024). One popular approach is to assess whether retrieval augmentation increases the likelihood of the LLM generating the correct answer and distill this observation to a supervised decision model (He et al., 2021; Schick et al., 2023; Xu et al., 2024). Analogously, Wang et al. (2023) and Wu et al. (2024) directly evaluate the correctness of the answers generated with and

without retrieval to create the supervision signal. Recent works have also explored solely examining the difficulty of a question to reveal the need for retrieval on question answering tasks (Mallen et al., 2023; Jeong et al., 2024; Asai et al., 2024). To further incorporate LLM's ability and confidence in the retrieval decision, Ding et al. (2024), Yao et al. (2024), and Moskvoretskii et al. (2025) use the LLM's uncertainty for selective retrieval. By contrast, this paper highlights the benefits of incorporating knowledge verbalization to make precise selective retrieval decisions and boosting the LLM's performance when retrieval is skipped.

Adaptive RAG Inference Strategies This paper relates to the field of adaptive RAG, which we define as designing configurable and instance-specific RAG inference strategies. Early works propose active retrieval, which drafts follow-up retrieval queries when the initially retrieved contexts are no longer relevant (Jiang et al., 2023; Su et al., 2024). Other studies investigate query decomposition and iterative retrieval, with the goal of decomposing complex queries into simpler ones to tackle (Shao et al., 2023; Kim et al., 2023; Liu et al., 2024; Lee et al., 2024). Given the retrieved knowledge, Asai et al. (2024) propose a tree search decoding algorithm supported by the ability to critique the retrieval and generation quality. Yan et al. (2024) retrieve from alternative sources to correct low quality retrievals. Parekh et al. (2025) incorporate an initial decision step to adaptively select the most suitable strategy based on the question. While this paper focuses on the more clearly defined selective retrieval problem, training an LLM to self-route between knowledge sources and act as a knowledge source itself represents a novel contribution to adaptive RAG as well.

LLMs as Knowledge Sources Using LLMs as knowledge generators has been extensively investigated by prior work. Shwartz et al. (2020) and Liu et al. (2022) utilize language models to generate background knowledge for unsupervised commonsense question answering. Yu et al. (2023) propose a general generate-and-read strategy that use LLMs as strong context generators for knowledge-intensive tasks. As another type of knowledge, a number of studies investigate using LLMs to generate intermediate steps to enhance reasoning (Wei et al., 2022; Kojima et al., 2022). Building upon these studies, this paper explores fully leveraging LLMs' knowledge generation ability to benefit selective retrieval systems.

3 Approach

In this section, we first reformulate selective retrieval as a knowledge source selection problem. Then, we introduce the details of the proposed Self-Routing RAG framework.

3.1 Problem Formulation

Knowledge Source-Aware Inference Given a user query q, a knowledge source s can be invoked to return related knowledge as a text sequence $s(q)$. Then, a reader LLM M generates the response $M(q, s(q))$. RAG implements this paradigm, instantiating s as a retriever that gathers relevant information from a specific external datastore.

Knowledge Source Selection In real world systems, there might be multiple available knowledge sources $\mathcal{S} \equiv \{\phi, s_1, ..., s_N\}$, where ϕ is a null knowledge source that returns nothing for any query. Based on the query q, a knowledge source selector P picks the most relevant source $P(q, \mathcal{S}) \in \mathcal{S}$ to query. Under this framework, selective retrieval can be expressed as $M(q, P(q, \{\phi, s\})(q))$ that optionally retrieves from a single external source s.

3.2 Self-Routing RAG: Overview

We propose Self-Routing RAG (SR-RAG), a selective retrieval framework that fully leverages the ability of LLMs as knowledge sources. As illustrated in Figure 1, based on the query q, the LLM autonomously determines which knowledge source to use—either retrieving from an external source or verbalizing its parametric knowledge. The final response is then generated based on both the query and the knowledge collected from the selected source.

Building upon traditional selective retrieval methods (Asai et al., 2024; Wu et al., 2024), SR-RAG fine-tunes the LLM to streamline its inference process with special tokens, enabling efficient inference with a single left-to-right generation pass. Three sets of special tokens are introduced:

1. A token <EOQ> marking the end of the query, which triggers knowledge source reflection[1].

[1]This design aligns with Wu et al. (2024) but diverges from Asai et al. (2024). We argue that this token is necessary for the LLM to allocate the probability mass to the tokens for the knowledge sources.

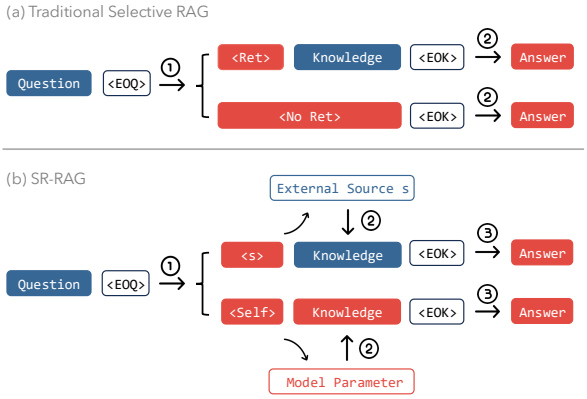

Figure 2: Compared with traditional selective RAG, SR-RAG triggers knowledge verbalization when retrieval is abstained. The LLM always generates the response conditioned on both the question and the knowledge. SR-RAG is also extensible to multiple output sources. We use blue to represent external information and red to represent the LLM and its self-generated tokens.

2. One special token <s> to represent each knowledge source s. In this paper, the main setting consists of two knowledge sources (one external and one internal). This formulation naturally accommodates additional knowledge sources as well.
3. A token <EOK> marking the end of the knowledge, which triggers answer generation.

As shown in Figure 2, compared to traditional selective retrieval, SR-RAG enables the LLM to actively select the best knowledge source and seamlessly act as a knowledge source itself.

3.3 Self-Routing RAG: Training

To train the backbone LLM for SR-RAG, we propose a pipeline that mines self-supervision from widely available question answering or instruction following data with pairs of question q and response a. Our pipeline only uses the LLM itself as knowledge source S_i and the external knowledge source S_e, without requiring additional human supervision or synthetic labels from stronger LLMs, demonstrating a strong scalability.

Data Construction To enable an LLM to accurately determine whether a question falls within its parametric knowledge and to robustly elicit the knowledge, we argue that thorough and diverse knowledge verbalization is crucial. Following this intuition, we collect contexts through rolling out with two knowledge sources:

- *Parametric Knowledge Verbalization*: We leverage GenRead (Yu et al., 2023) to elicit knowledge from the LLM parametric knowledge source S_i and generate n diverse verbalized contexts, denoted as $c_{i_1}, c_{i_2}, ..., c_{i_n}$.
- *External Knowledge Retrieval*: We retrieve n context chunks from the external knowledge source S_e, denoted as $c_{e_1}, c_{e_2}, ..., c_{e_n}$. In this work, we consider retrieving from Wikipedia with an off-the-shelf dense retriever.

For each context $c_j \in \{c_{i_1}, ..., c_{i_n}, c_{e_1}, ..., c_{e_n}\}$, we measure their helpfulness as the likelihood of the LLM generating a as $l_j = p_M(a|q, c_j)$. After sorting l_j, we identify the preferred source $s \in \{S_i, S_e\}$ as the one contributing to the majority of top-n ranked helpful knowledge items. The data creation pipeline then outputs the tuple $(q, a, s, \{c_j, l_j\})$ for model training. For convenience of later reference, we further denote the contexts from S_i and S_e that lead to the highest and lowest likelihoods as $c_{i+}, c_{i-}, c_{e+}, c_{e-}$, respectively. We present the formal data creation algorithm in Appendix B.1.

Objective SR-RAG proposes a two-stage multi-task learning framework that jointly optimizes knowledge source selection, knowledge verbalization, and response generation. The first stage performs behavior cloning on three losses:

1. $\mathcal{L}_{src}$: a cross-entropy loss for the preferred knowledge source s following <EOQ>:

$$\mathcal{L}_{src} = -\log p_M(\texttt{<s>}|q), \tag{1}$$

where <s> represents the actual token corresponding to the chosen source $s \in \mathcal{S}$.

2. $\mathcal{L}_{verb}$: a cross-entropy loss on the knowledge tokens, only when the LLM itself (S_i) is labeled as the preferred knowledge source:

$$\mathcal{L}_{verb} = \begin{cases} -\log p_M(c_{i+}|q), & \text{if } s = S_i, \\ 0, & \text{if } s = S_e. \end{cases} \tag{2}$$

3. $\mathcal{L}_{ans}$: a cross-entropy loss on generating the answer based on q and the knowledge:

$$\mathcal{L}_{ans} = \begin{cases} -\log p_M(a|q, c_{i+}), & \text{if } s = S_i, \\ -\log p_M(a|q, c_{e+}), & \text{if } s = S_e. \end{cases} \tag{3}$$

The final loss for the first stage is a simple combination of the three objectives:

$$\mathcal{L}_{stage1} = \mathcal{L}_{src} + \mathcal{L}_{verb} + \mathcal{L}_{ans}. \quad (4)$$

To further boost the LLM's ability to generate useful knowledge, SR-RAG incorporates a second-stage fine-tuning via direct preference optimization (DPO) (Rafailov et al., 2023), pairing self-verbalized knowledge with self-generated preference labels (c_{i+}, c_{i-}).

$$\mathcal{L}_{stage2} = \mathcal{L}_{src} + \mathcal{L}^{DPO}_{verb} + \mathcal{L}_{ans}, \quad (5)$$

$$\mathcal{L}^{DPO}_{verb} = \begin{cases} -\log\sigma\Big(\beta\log\frac{p_M(c_{i+}|q)}{p_{ref}(c_{i+}|q)} - \beta\log\frac{p_M(c_{i-}|q)}{p_{ref}(c_{i-}|q)}\Big), & \text{if } s = S_i, \\ 0, & \text{if } s = S_e. \end{cases} \quad (6)$$

M and ref are initialized with the LLM fine-tuned on $\mathcal{L}_{stage1}$, and only M is updated.

Overall, this self-supervised pipeline effectively binds knowledge verbalization with the selective retrieval paradigm. The LLM learns accurate knowledge source preferences through performance-oriented labeling. Furthermore, analogous to distilling complex "System 2" reasoning into fast "System 1" inference (Yu et al., 2024), the LLM learns from high-quality knowledge that was computationally expensive to collect to perform cost-efficient knowledge verbalization at inference time. Finally, SR-RAG naturally extends to more than two knowledge sources, which useful for training the LLM to further distinguish domain-specific corpora or retrieval methods that have different cost-quality trade-offs.

3.4 Self-Routing RAG: Inference

As shown in Figure 1 and Figure 2, SR-RAG inference streamlines three steps in a single left-to-right pass: Source Selection, Knowledge Collection, and Answer Generation.

Retrieval-Augmented Source Selection A common approach to selecting a knowledge source is to compare the likelihood $p_M(\texttt{<s>}|q)$ for each $s \in \mathcal{S}$, against a fixed threshold (Asai et al., 2024; Wu et al., 2024). However, this approach does not account for shifts in the LLM's ability after fine-tuning and does not fine-grained control over the decision boundary. To make source selection more robust, we propose a nearest neighbor-based selection mechanism that dynamically adapts to different inputs. Concretely, the fine-tuned LLM is evaluated on a set of question-answer pairs[2]. The probabilities of generating the answer a conditioned on different knowledge sources s are compared to decide the preferred one. Then, a policy datastore is constructed to map q to its preferred source. The hidden representation of <EOQ> is used as the key. At test time, we retrieve k nearest neighbors from the policy datastore and use their source labels to form a distribution over the sources $p_D(\texttt{<s>}|q)$. Finally, to select the best source $s \in \mathcal{S}$, we apply a threshold on the product

$$p_M(\texttt{<s>}|q) \times p_D(\texttt{<s>}|q). \quad (7)$$

While tackling the challenges to source selection due to the LLM's ability shift, this approach also exhibits a better interpretability: since the policy datastore consists of explicit source assignments, it can be audited, modified, and expanded by human experts to further improve SR-RAG's source selection performance in different domains.

Subsequently, the knowledge from the corresponding source is gathered. If the LLM prefers S_i, we use greedy decoding to directly verbalize a single knowledge context. After SR-RAG fine-tuning, the generated context serves as a compressed yet high-quality articulation of the parametric knowledge, which would otherwise require compute-expensive knowledge verbalization to elicit. If the LLM instead selects an external source <s>, we halt decoding and retrieve from s. With the retrieved or verbalized knowledge appended to context, the LLM proceeds to generate the final response.

4 Experimental Setup

4.1 SR-RAG Implementation Details

Data Construction We experiment SR-RAG with a set of two knowledge sources: the 2018 English Wikipedia (<Wiki>) as the external knowledge source, and the LLM itself (<Self>) as the internal knowledge source. For the Wikipedia, we use the official embeddings released by Karpukhin et al. (2020) and retrieve in the granularity of 100-word chunks. For the LLM itself, we use GenRead (Yu et al., 2023) to verbalize diverse knowledge contexts during training data construction. GenRead first clusters zero-shot knowledge verbalizations in the same domain and uses them as in-context demonstrations to verbalize diverse

[2]Our experiments reuse the SR-RAG training set so that no additional supervision is required.

knowledge. We limit each verbalized knowledge chunk to maximum 150 tokens. For each knowledge source, we collect $n = 5$ knowledge chunks.

Training We fine-tune Llama-2-7B-Chat released by Touvron et al. (2023) on a mixture of six short-form and long-form knowledge-intensive datasets: Wizard of Wikipedia (Dinan et al., 2019), Natural Questions (Kwiatkowski et al., 2019), FEVER (Thorne et al., 2018), OpenBookQA (Mihaylov et al., 2018), ARC-Easy (Bhakthavatsalam et al., 2021), and ASQA (Stelmakh et al., 2022). This mixture of 53,042 instances is a subset of the RAG-oriented instruction tuning data proven effective in Asai et al. (2024). After running the data construction algorithm discussed in §3.3, 30.7% of the instances are labeled with `<Self>` and the rest are labeled with `<Wiki>` as the preferred knowledge source. We present further details regarding the training data in Appendix B.1. For stage 1 training, we use batch size 64, learning rate 1e-5, and fine-tune for 1 epoch. For stage 2 training, we use batch size 64, learning rate 5e-7, $\beta = 0.3$ for DPO, and train for another epoch. All the experiments are performed on a local machine with eight A800 (80GB) GPUs and a local machine with eight A6000 GPUs. On eight A800 (80GB) GPUs, the two-staged training takes approximately 10 hours.

Inference To construct the policy datastore, we use a middle layer in the fine-tuned LLM[3] as middle layers are found effective by previous work on LLM faithfulness (Yin et al., 2024). At test time, the datastore index is cached on GPU and similarity search can be achieved via a single matrix multiplication. We retrieve $k = 30$ nearest supporting examples from the datastore andconstruct $p_D(\texttt{<s>}|q)$ from the counts of each knowledge source as the preferred source. Then, we impose a model-specific threshold τ on $p_M(\texttt{<Wiki>}|q) \times p_D(\texttt{<Wiki>}|q)$ to decide whether retrieval should be triggered[4]. We find that this threshold generally performs well enough and does not require dataset-specific tuning.

4.2 Evaluation

Datasets and Metrics We test SR-RAG on a diverse set of four knowledge-intensive NLP tasks:

- **PopQA** (Mallen et al., 2023): a free-formed long-tail open-domain question answering dataset. Following Asai et al. (2024), we use the subset of 1,399 questions that aims at testing long-tail entities.

- **TriviaQA** (Joshi et al., 2017): an established open-domain question answering dataset that features relatively complex and diverse questions. We use the same test split as in Asai et al. (2024).

- **PubHealth** (Zhang et al., 2023): a fact-checking dataset focusing on public health-related claims.

- **ARC Challenge** (Bhakthavatsalam et al., 2021): a multiple-choice question answering dataset featuring grade-school level science questions.

Following common practice, we perform lexical postprocessing of the model's output and report accuracy for PubHealth and ARC and substring matching for PopQA and TriviaQA.

Baselines We compare SR-RAG with the following baselines that incorporate comprehensive training and inference strategies: (1) First, we present comparisons with baselines that either always retrieve or always verbalize using the original model before fine-tuning. In this setting, *GenRead* is used as the verbalization method. (2) As illustrated in Figure 2, the major baseline we compare SR-RAG with is the state-of-the-art prior *selective retrieval* pipeline, combining the advantage of He et al. (2021), Asai et al. (2024), and Wu et al. (2024). Specifically, the likelihoods of the LLM generating the answer with and without retrieval are used to create the knowledge source selection label. Then, we fine-tune the LLM for knowledge source selection (among S_e and ϕ) and generate the answer with optional retrieval. At inference time, we apply a uniform threshold of 0.2 on the likelihood of the retrieval token following `<EOQ>` for selectively triggering retrieval. Besides, we also provide the result of always retrieving at inference time. (3) Always retrieving with the fine-tuned SR-RAG model.

[3] Layer 15 for Llama-2-7B-Chat and Phi-3.5–Mini-Instruct and layer 11 for Qwen2.5-7B-Instruct.

[4] $\tau = 0.1$ for Llama-2-7B-Chat and $\tau = 0.2$ for the other two LLMs.

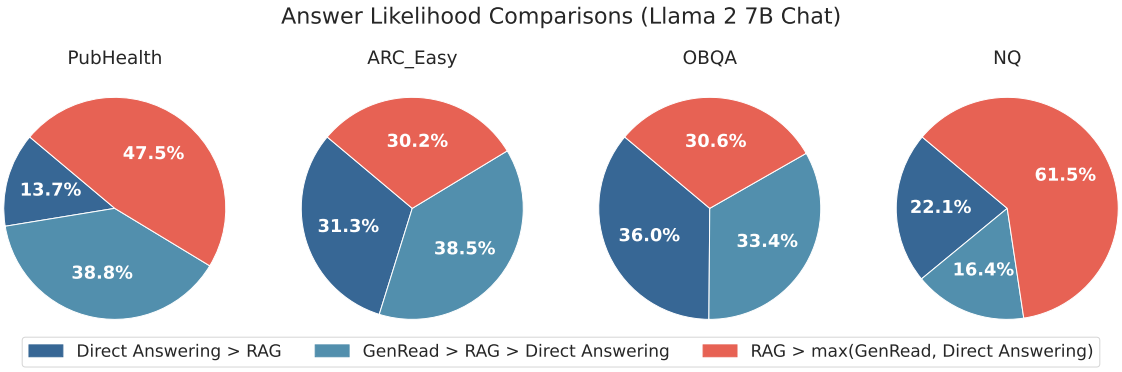

Figure 3: Knowledge verbalization significantly affects the LLM ability boundary measured for selective retrieval.

Training	Inference	PopQA		TriviaQA		PubHealth		ARC		Average	
		ACC	%RAG	ACC	%RAG	ACC	%RAG	ACC	%RAG	ACC	%RAG
Llama-2-7B-Chat											
No Fine-tuning	Always RAG	0.529	100%	0.641	100%	0.457	100%	0.546	100%	0.543	100%
	GenRead	0.247	0%	0.616	0%	0.515	0%	0.605	0%	0.496	0%
Selective RAG	Always RAG	0.567	100%	0.640	100%	0.588	100%	0.588	100%	0.596	100%
	Selective RAG	0.565	98%	0.638	100%	0.589	100%	0.594	65%	0.597	86%
SR-RAG	Always RAG	**0.568**	100%	**0.669**	100%	0.689	100%	0.608	100%	0.634	100%
	SR-RAG	0.566	96%	0.664	89%	**0.715**	40%	**0.630**	29%	**0.644**	64%
Phi-3.5-Mini-Instruct											
No Fine-tuning	Always RAG	0.541	100%	0.594	100%	0.549	100%	0.771	100%	0.614	100%
	GenRead	0.331	0%	0.567	0%	0.442	0%	0.840	0%	0.545	0%
Selective RAG	Always RAG	**0.570**	100%	0.645	100%	0.701	100%	0.813	100%	0.682	100%
	Selective RAG	**0.570**	100%	0.638	95%	0.704	91%	0.815	83%	0.682	92%
SR-RAG	Always RAG	0.567	100%	**0.659**	100%	0.689	100%	0.820	100%	0.684	100%
	SR-RAG	0.566	98%	0.657	92%	**0.705**	24%	**0.854**	5%	**0.696**	55%
Qwen-2.5-7B-Instruct											
No Fine-tuning	Always RAG	0.563	100%	**0.667**	100%	0.446	100%	**0.916**	100%	0.648	100%
	GenRead	0.334	0%	0.626	0%	0.676	0%	0.875	0%	0.628	0%
Selective RAG	Always RAG	0.555	100%	0.654	100%	0.600	100%	0.827	100%	0.659	100%
	Selective RAG	0.529	88%	0.648	93%	0.608	82%	0.835	78%	0.655	85%
SR-RAG	Always RAG	**0.573**	100%	0.662	100%	0.596	100%	0.821	100%	0.663	100%
	SR-RAG	0.572	99%	0.659	89%	**0.682**	34%	0.830	46%	**0.686**	67%

Table 1: Main evaluation results on four tasks. The best results are boldfaced. SR-RAG significantly outperforms selective RAG and always retrieving while requiring a much lower retrieval budget.

5 Results

5.1 Knowledge Verbalization Impacts Knowledge Source Choices

As a proof of concept, we first show that on a range of knowledge-intensive tasks, knowledge verbalization leads to significantly different interpretations of the LLM's ability boundary, which in turn significantly affects the preference labeling process of selective retrieval. As shown in Figure 3, we use four datasets in SR-RAG's train set mixture and report the likelihood of the model generating the correct answer given no context (dark blue), the most helpful GenRead context (c_{i+}, light blue), and the most helpful retrieved context (c_{e+}, red). Surprisingly, GenRead reverses the knowledge source preference on 16% instances from Natural Questions and more than 30% on the other three datasets. This result highlights that prior selective retrieval methods may significantly underestimate the ability of the LLM, further motivating the necessity of embracing knowledge verbalization for

Method	PopQA	TriviaQA	PubHealth	ARC	Average
Accuracy (Verbalization ≥ Retrieval)					
Self-RAG	0.957	0.936	0.867	0.908	0.917
SR-RAG w/o. kNN	**0.959**	0.930	0.869	0.888	0.912
SR-RAG	**0.959**	**0.943**	**0.880**	**0.910**	**0.923**
AUROC (Retrieval > Verbalization)					
Self-RAG	0.489	0.503	0.438	**0.557**	0.497
SR-RAG w/o. kNN	0.490	**0.567**	0.564	0.513	0.534
SR-RAG	**0.577**	0.565	**0.606**	0.533	**0.570**

Table 2: Source selection accuracy of different strategies. SR-RAG achieves the highest average accuracy and AUROC in two test scenarios.

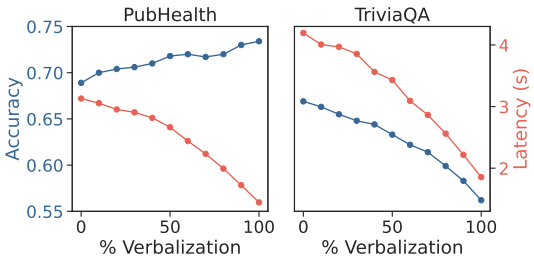

Figure 4: Latency-Accuracy trade-off of SR-RAG with different verbalization frequencies.

accurately labeling knowledge source preferences.

5.2 Overall Generation Performance

Table 1 shows the end-to-end generation performance on three LLMs, demonstrating the advantage of SR-RAG over selective retrieval and other baselines. Specifically, although training the model with the baseline selective retrieval saves 8% to 15% retrieval, its final generation performance is nearly identical to always retrieving. This indicates that selective RAG cannot offer an effective retrieval-skipping strategy and cannot improve the overall performance over always RAG. In comparison, SR-RAG outperforms always RAG by skipping 20% to 40% of partial low-quality retrievals and effectively verbalizing parametric knowledge. Remarkably, with a uniform inference datastore and threshold, SR-RAG is able to dynamically adapt to the difficulty of the datasets. For a dataset that emphasizes long-tail knowledge like PopQA, SR-RAG tends to retrieve external knowledge most of the time. On the other hand, for PubHealth and ARC where the model's knowledge may suffice for a number of questions, SR-RAG relies on internal knowledge more confidently, resulting in much better performance compared to always retrieving.

5.3 Source Selection Performance

Can SR-RAG accurately select its knowledge source? In Table 2, we compare SR-RAG with Self-RAG and SR-RAG without kNN based on two accuracy definitions: (1) not harming the performance when abstaining retrieval (top) and (2) selecting retrieval only when it is strictly greater than verbalization (bottom). In the former setting, despite all methods have a relatively high performance, SR-RAG demonstrates the best average accuracy. However, in the latter setting, SR-RAG clearly demonstrates better discrimination ability, outperforming Self-RAG by 14.7% in the average AUROC. Removing the nearest neighbor source selection strategy (w/o. kNN), we find both averaged Accuracy and AUROC dropped significantly, indicating its advantage in adapting to the LLM ability shift caused by fine-tuning.

5.4 System Efficiency

We further measure the end-to-end system latency of SR-RAG with Llama-2-7B-Chat in a batched inference setting[5]. In Figure 4, we illustrate how accuracy and latency vary with respect to different proportions of cases in the PubHealth and TrivialQA datasets selected for knowledge verbalization. We can consistently conclude that latency improves as the proportion of verbalization increases. This indicates that reducing low-quality retrieval and eliciting the inherent knowledge from the LLM itself can significantly improve system efficiency. Due to their varied difficulty, the best accuracy for different datasets is achieved at different verbalization proportions. However, as shown in Table 1, due to its advanced inference strategy, SR-RAG does not require specific tuning to achieve this optimality.

5.5 Ablation Study

To further verify the effectiveness of key components of SR-RAG, we conduct ablation studies on Llama-2-7B-Chat and present the results in Table 3. First, the dynamic policy inference via kNN is removed (w/o. kNN). The decrease in performance and increase in the retrieval proportion which corresponds to a higher budget illustrate that the kNN inference policy better adapts to the fine-tuned model's ability. In addition, we remove knowledge

[5]We provide the detailed latency model in Appendix B.2. We also show there that although latency is reported here, it is linear to the retrieval frequency when the batch size is small.

Method	PopQA	TriviaQA	PubHealth	ARC	Average
Answer Accuracy (↑)					
SR-RAG	0.566	**0.664**	**0.715**	**0.630**	**0.644**
w/o. kNN	0.558	0.658	0.694	0.627	0.634
w/o. kv. label	**0.568**	0.644	0.598	0.629	0.610
w/o. $\mathcal{L}_{verb}^{DPO}$	0.564	0.645	0.674	0.581	0.616
%Retrieval (↓)					
SR-RAG	96%	89%	**40%**	**29%**	**64%**
w/o. kNN	**94%**	**77%**	72%	56%	75%
w/o. kv. label	100%	100%	100%	84%	96%
w/o. $\mathcal{L}_{verb}^{DPO}$	98%	100%	79%	66%	86%

Table 3: Ablation studies on SR-RAG.

verbalization for knowledge source labeling and use the likelihood of the LLm directly generating the answer as the baseline instead (w/o. kv. label). The model converges to over-relying on retrieval, which does not yield the best performance. Finally, $\mathcal{L}_{stage2}$ is ablated and the $\mathcal{L}_{stage1}$ loss is kept for stage 2 training. The result shows a significant increase in the retrieval proportion as the model's knowledge verbalization ability degrades.

6 Conclusion

We introduce SR-RAG, a novel RAG framework demonstrating that selective retrieval can be substantially improved by binding with knowledge verbalization. SR-RAG's training paradigm enables an LLM to accurately avoid retrieval and effectively verbalize its own knowledge. At inference stage, SR-RAG leverages the model's internal states to improve its selective retrieval accuracy. Extensive evaluations demonstrate our approach's effectiveness in enhancing accuracy while significantly reducing latency, paving the way to more reliable and adaptive RAG systems.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *ArXiv preprint*, abs/2404.14219.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Physics of language models: Part 3.2, knowledge manipulation. *CoRR*, abs/2309.14402.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Sumithra Bhakthavatsalam, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, and Peter Clark. 2021. Think you have solved direct-answer question answering? try arc-da, the direct-answer AI2 reasoning challenge. *CoRR*, abs/2102.03315.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, and Xueqi Cheng. 2024. Retrieve only when it needs: Adaptive retrieval augmentation for hallucination mitigation in large language models. *CoRR*, abs/2402.10612.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021. Efficient nearest neighbor language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5703–5714, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Zhengbao Jiang, Frank 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, Singapore. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale

distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur 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 Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Myeonghwa Lee, Seonho An, and Min-Soo Kim. 2024. PlanRAG: A plan-then-retrieval augmented generation for generative large language models as decision makers. 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 6537–6555, Mexico City, Mexico. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169, Dublin, Ireland. Association for Computational Linguistics.

Yanming Liu, Xinyue Peng, Xuhong Zhang, Weihao Liu, Jianwei Yin, Jiannan Cao, and Tianyu Du. 2024. RA-ISF: Learning to answer and understand from retrieval augmentation via iterative self-feedback. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4730–4749, Bangkok, Thailand. 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.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Viktor Moskvoretskii, Maria Lysyuk, Mikhail Salnikov, Nikolay Ivanov, Sergey Pletenev, Daria Galimzianova, Nikita Krayko, Vasily Konovalov, Irina Nikishina, and Alexander Panchenko. 2025. Adaptive retrieval without self-knowledge? bringing uncertainty back home.

Tanmay Parekh, Pradyot Prakash, Alexander Radovic, Akshay Shekher, and Denis Savenkov. 2025. Dynamic strategy planning for efficient question answering with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

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.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

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, 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.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4615–4629, Online. Association for Computational Linguistics.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12991–13013, Bangkok, Thailand. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288.

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023. Self-knowledge guided retrieval augmentation for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10303–10315, Singapore. Association for Computational Linguistics.

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 *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Di Wu, Wasi Uddin Ahmad, Dejiao Zhang, Murali Krishna Ramanathan, and Xiaofei Ma. 2024. Repoformer: Selective retrieval for repository-level code completion. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024. RECOMP: improving retrieval-augmented lms with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *ArXiv preprint*, abs/2412.15115.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Weichuan Liu, Lei Hou, and Juanzi Li. 2024. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation. *CoRR*, abs/2406.19215.

Fan Yin, Jayanth Srinivasa, and Kai-Wei Chang. 2024. Characterizing truthfulness in large language model generations with local intrinsic dimension. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024. Distilling system 2 into system 1. *CoRR*, abs/2407.06023.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023.

Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Tianhua Zhang, Hongyin Luo, Yung-Sung Chuang, Wei Fang, Luc Gaitskell, Thomas Hartvigsen, Xixin Wu, Danny Fox, Helen Meng, and James R. Glass. 2023. Interpretable unified language checking. *CoRR*, abs/2304.03728.

Supplementary Material: Appendices

A List of notations

In Table 4, we present the major notations and parameters used throughout the paper.

Notation	Description
q	User query input to the system.
a	The expected answer.
M	The LLM.
$\mathcal{S}$	Set of all knowledge sources.
S_e	The external knowledge source.
S_i	The internal knowledge source (parametric knowledge).
c_{i+}	Most helpful verbalized knowledge context from S_i.
c_{i-}	Least helpful verbalized knowledge context from S_i.
c_{e+}	Most helpful retrieved knowledge context from S_e.
c_{e-}	Least helpful retrieved knowledge context from S_e.
<EOQ>	End-of-query special token.
<EOK>	End-of-knowledge special token.
<s>	Special token representing knowledge source $s \in \mathcal{S}$.
<Wiki>	Special token representing Wikipedia.
<Self>	Special token representing LLM as knowlede source.
k	number of neighbors retrieved for source policy inference

Table 4: A summary of the key symbols and parameters used in the paper.

B SR-RAG: Further Details

B.1 Training Details

Dataset Construction Algorithm We present the full algorithm that we use to construct the training data and the preference labels for SR-RAG in Algorithm 1. Note that GenRead is run separately for each training data subset. We use instance-level notations for better readability.

Algorithm 1 SR-RAG Training Data Construction

Require: LLM M, External Retriever $\mathcal{R}$, Dataset $\mathcal{D}$, Number of contexts n

for $(q, a) \in \mathcal{D}$ **do**
  // Retrieving External Knowledge
  $\mathcal{C}_r \leftarrow \mathcal{R}(q, n)$
  // Knowledge Verbalization
  $\mathcal{C}_v \leftarrow M$.GenRead$(q, n)$
  // Compute Likelihoods
  for $c \in \mathcal{C}_r \cup \mathcal{C}_v$ **do**
    $l_c \leftarrow p_M(a|q, c)$
  end for
  $s \leftarrow \arg\max_{s \in \{S_i, S_e\}} \sum_{c \in \mathcal{C}_s} l_c$
  Store $(q, a, s, \{c_i, l_i\})$
end for
return Processed dataset with labeled knowledge sources

GenRead Prompt We closely follow the original paper (Yu et al., 2023) to implement GenRead. For in-context examples in round 2 verbalization, we use five clusters and five example from each cluster. For both rounds of verbalization, we use the following prompt as shown in Figure 5 for all the datasets except ASQA. For ASQA, we add the statement "`If the question is ambiguous, generate multiple documents for each possibility`" to instruct the model consider the potential abiguity in generating the background knowledge.

```
Generate a background document from Wikipedia
to help answer the following question. Directly
start with document content and do not generate
URL.

Question: {question}

Background document:
```

Figure 5: Prompt used for knowledge verbalization data collection via GenRead.

Training data details In Table 5, we present the detail of the train and validation data for SR-RAG. We also provide the percentage of instances where verbalization is preferred over retrieval.

Dataset	Train	Validation	Total	%Verbalization		
				Llama	Phi	Qwen
ARC_Easy	2037	107	2144	61%	84%	66%
NQ	14753	776	15529	28%	33%	41%
OBQA	4462	234	4696	61%	77%	61%
FEVER	9467	498	9965	52%	58%	68%
WoW	16493	868	17361	13%	55%	32%
ASQA	3700	194	3894	13%	25%	16%

Table 5: Statistics of training and validation data with verbalization percentages. Llama = Llama-2-7B-Chat, Phi = Phi-3.5-Mini-Instruct and Qwen = Qwen2.5-7B-Instruct.

B.2 Formulation for Latency Experiments

To evaluate the inference efficiency of SR-RAG, we measure the latency under a realistic batched inference setup, where the system handles a batch of B = 10 queries and returns the results for all of them together. We choose this setup due to the complexity of the retrieval system. For instance, in our implementation of Wikipedia search, it takes around 10 seconds for encoding the query and retrieving the most relevant context chunks. For the latency experiments presented in the paper, we assume that the retrieval index is pre-constructed, and define the following latency components:

- **Source Selection Time** (T_d): The time taken by the knowledge source selector to determine whether to retrieve from external sources or rely on parametric knowledge.
- **Retrieval Latency** (T_r): If retrieval is triggered, the time taken to fetch external knowledge from the database. In our batched setting, we calculate T_r by performing a batched retrieval for all the instances that require retrieval and use the report a per-item latency.
- **Verbalization Latency** (T_v): If retrieval is not triggered, the time taken for the model to verbalize parametric knowledge.
- **Generation Latency** (T_g): The time required for the language model to generate the response, conditioned on either retrieved or verbalized knowledge.

Thus, the total per-item latency T_{total} is given by:

$$T_{\text{total}} = \begin{cases} T_d + T_v + T_g, & \text{if verbalize} \\ T_d + T_r + T_g, & \text{if retrieve} \end{cases}$$

Generally, if the batch size is small, it is safe to assume that $T_r >> T_v \approx T_g >> T_d$. In the extreme online setting, the system's efficiency gain converges to the percentage of retrieval avoided.