Self-Routing 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 aims to make retrieval-augmented generation (RAG) more efficient and reliable by skipping retrieval when an LLM's parametric knowledge suffices. Despite promising results, existing methods are constrained by a binary design choice: either retrieve from a single external source or skip retrieval and let the LLM directly produce the final answer. We argue that this fallback underestimates the model's knowledge and obscures the more general multi-source decision problem that arises in practical systems. We propose **Self-Routing RAG (SR-RAG)**, which casts selective retrieval as knowledge source selection and treats the LLM itself as a first-class knowledge source. SR-RAG learns to select an appropriate knowledge source, optionally verbalize parametric knowledge, and answer using the selected source, all within a single left-to-right generation pass. SR-RAG further augments source selection by combining LLM-based uncertainty with a flexible external policy datastore to improve decision calibration. Across four benchmarks and three 7B-class LLMs, SR-RAG outperforms a strong selective retrieval baseline by 8.5%/2.1%/4.7% while performing 26%/40%/21% fewer retrievals, and it achieves favorable accuracy-latency trade-offs without dataset-specific threshold tuning.

1 Introduction

Retrieval-augmented generation (RAG) equips large language models (LLMs) with external knowledge at inference time, improving performance on tasks that require domain-specific or up-to-date knowledge (Khandelwal et al., 2020; Lewis et al., 2020; Borgeaud et al., 2022; Ram et al., 2023; Shi et al., 2024). However, retrieval can be unnecessary when retrieved passages overlap with the LLM's parametric knowledge, or even harmful when the passages are noisy or irrelevant. These limitations motivate *selective retrieval*, where the system retrieves only when retrieval is likely to be useful (He et al., 2021; Mallen et al., 2023; Xu et al., 2024; Asai et al., 2024; Wu et al., 2024a).

A limitation of current selective retrieval methods is reducing selective retrieval to a binary decision between consulting a single external source and a no-retrieval fallback where the LLM directly generates the final answer (Mallen et al., 2023; Jeong et al., 2024; Asai et al., 2024; Wu et al., 2024a). This design provides a coarse characterization of what the model can do without retrieval, as direct answer generation removes an opportunity to explicitly surface relevant parametric knowledge before committing to an answer. Consequently, routers trained by comparing RAG against direct answering, either via answer correctness (Wang et al., 2023; Wu et al., 2024a) or likelihood (He et al., 2021; Xu et al., 2024), can learn preferences inconsistent with the LLM's non-retrieval capability. We observe this in a pilot study, where eliciting and conditioning on verbalized knowledge changes the preferred source for a substantial fraction of instances (§5.1). Furthermore, the binary setup is inflexible as it cannot learn to leverage multiple knowledge sources, which could be an important limitation in real-world deployments.

We propose **Self-Routing RAG (SR-RAG)**, which addresses both limitations by casting selective retrieval as a *knowledge source selection* problem and treating the LLM itself as a first-class knowledge source. SR-RAG expands the selective retrieval action space to routing between an internal knowledge source and an expandable set of external sources. As shown in Figure 1, at inference time, when the internal source is selected, the LLM first *verbalizes* query-relevant parametric knowledge as a compact context, then answers conditioned on it. When an external source is selected, decoding is paused to retrieve evidence from the selected index, and the model answers conditioned

*Equal Contribution

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 524–545
November 6-10, 2026 ©2026 Association for Computational Linguistics

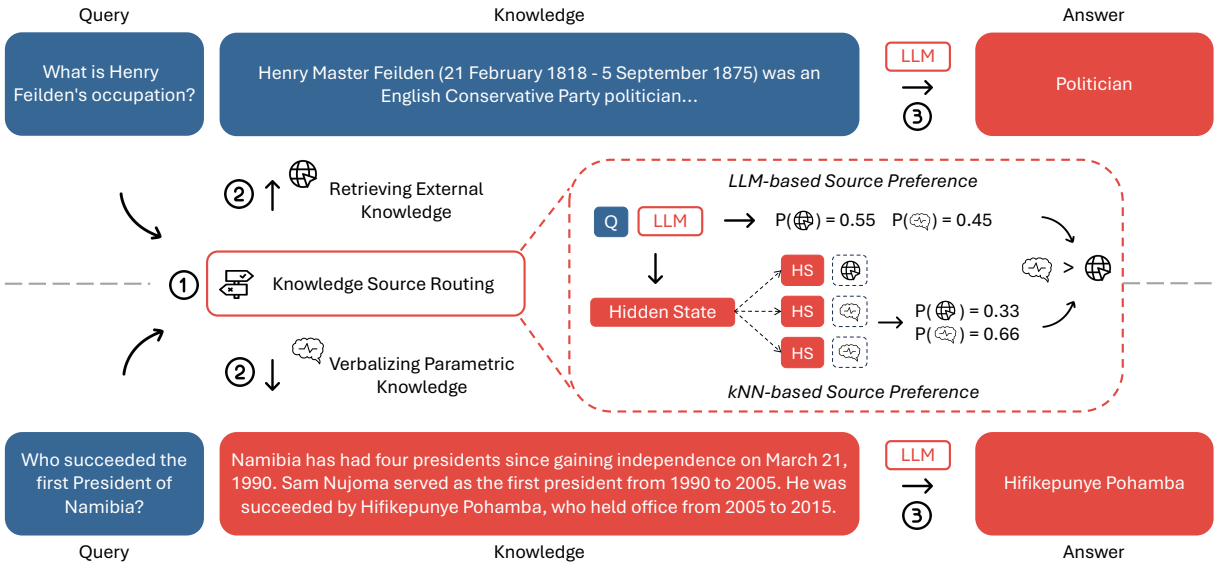


Figure 1: **Self-Routing RAG Inference.** (1) A knowledge source is selected based on a query by combining the LLM's likelihood of a special source token with neighbor decisions from an external policy datastore. (2) If an external source is selected, the passages are retrieved and appended as context. If the internal source is selected, the LLM generates a compact verbalized knowledge context. (3) The LLM then produces the final answer conditioned on the query and the selected knowledge. All steps can be executed in a single left-to-right generation pass.

on the retrieved passages.

To make SR-RAG reliable and efficient, we propose three techniques that span data construction, training, and inference. **(1) Verbalization-aware supervision:** given each training question, we collect multiple candidate contexts from each source, including diverse self-verbalized knowledge paragraphs, and assign the preferred source as the one whose contexts most increase the likelihood of the gold answer. This produces source selection labels that better reflect the LLM's capability without retrieval than direct-answer baselines. **(2) Multi-task learning:** we jointly train an LLM as a better knowledge source router, a better knowledge source, and a better knowledge reader in RAG. A two-staged multi-task objective is proposed that couples source selection, knowledge verbalization, and answer generation, including a preference alignment training to further boost the knowledge verbalization ability. **(3) kNN-augmented source selection:** purely relying on LLM-based uncertainty to select a source as done in previous work is brittle and agnostic to training-induced LLM ability shifts. We propose to build a policy datastore from light rollouts and combine the model's source selection probabilities with neighbor-defined source distributions in the hidden state space at test time.

We evaluate SR-RAG by fine-tuning 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 tasks and testing on four benchmarks. SR-RAG outperforms a strong standard selective retrieval baseline by 8.5%/2.1%/4.7% while performing 26%/40%/21% fewer retrievals across the three LLMs, respectively (§5.2). We further show that SR-RAG makes more accurate source selection decisions (§5.3) and achieves favorable accuracy-latency trade-offs without dataset-specific threshold tuning (§5.4, Figure 4). Ablations confirm that verbalization-aware labeling, preference alignment, and kNN-augmented inference each contribute materially to these gains (§5.5). Finally, we experiment on routing among three sources (LLM, Wikipedia, and PubMed), where SR-RAG learns to retrieve from specialized corpus when needed while retaining strong performance on general workloads (§B). Data and code will be publicly released at `https://github.com/xiaowu0162/self-routing-rag`.

2 Related Work

Selective Retrieval To enhance the efficiency of RAG systems and reduce harmful or distracting retrievals, several works propose selectively skipping retrieval augmentation (He et al., 2021; Mallen et al., 2023; Xu et al., 2024; Wu et al., 2024a). A common approach assesses whether retrieval in-

creases the likelihood of generating the correct answer and distills this signal into a supervised decision model (He et al., 2021; Schick et al., 2023; Xu et al., 2024). Similarly, Wang et al. (2023) and Wu et al. (2024a) compare answers generated with and without retrieval to form supervision based on correctness. Other lines of work examine the question alone to decide whether retrieval is needed (Mallen et al., 2023; Jeong et al., 2024; Asai et al., 2024). To incorporate model confidence, recent methods use uncertainty-based selective retrieval (Ding et al., 2024; Yao et al., 2024; Moskvoretskii et al., 2025a). In contrast to prior work that treats skipping retrieval as direct answer generation, our work highlights that routing decisions and non-retrieval performance both improve when the LLM is treated as a knowledge source and is trained to verbalize query-relevant parametric knowledge.

Adaptive RAG Inference This work also relates to adaptive RAG, which studies instance-specific inference strategies beyond a fixed retrieve-then-read pipeline. Prior work studies *active retrieval*, where the system re-issues queries when initial retrieval is insufficient (Jiang et al., 2023; Su et al., 2024). Another line explores query decomposition and iterative retrieval to handle complex questions by breaking them into sub-queries (Shao et al., 2023; Kim et al., 2023; Liu et al., 2024; Lee et al., 2024). Given retrieved results, Asai et al. (2024) and Yan et al. (2024) critique or revise the retrieved knowledge to improve output quality. Parekh et al. (2025) incorporate an initial decision step to adaptively select the most suitable strategy based on the question. While SR-RAG focuses on the targeted decision of which knowledge source to consult, it introduces a complementary form of adaptivity by allowing the model to route among heterogeneous sources, including its own parametric knowledge, within a single generation pass.

LLMs as Knowledge Sources A growing body of work explores using LLMs to generate auxiliary knowledge. Early studies show that LLMs can produce relevant background knowledge in zero-shot settings for commonsense reasoning (Shwartz et al., 2020; Liu et al., 2022). More broadly, Yu et al. (2023) propose a generate-then-read approach, using the LLM as a context generator in place of external retrieval for RAG. LLMs have also been shown to generate effective intermediate reasoning steps for complex questions (Wei et al., 2022; Kojima et al., 2022). Building on these insights, our work integrates LLM-generated knowledge into selective retrieval by making it an explicit, routable source, and by training the model to both decide when to rely on it and to produce compact verbalizations that improve end-to-end answering.

3 Approach

We first reformulate selective retrieval as knowledge source selection. We then present Self-Routing RAG (SR-RAG), a framework that allows an LLM to self-route between external retrieval and internal knowledge verbalization within a single left-to-right generation pass.

3.1 Problem Formulation

Knowledge Source-Aware Inference Given a user query q, a knowledge source s returns relevant information as a text sequence $s(q)$ (possibly empty). An LLM reader M then produces a response conditioned on both the query and the returned knowledge:

$$M(q, s(q)). \tag{1}$$

Under this framework, standard RAG is an instantiation with s being a retriever over an external datastore and $s(q)$ being retrieved passages.

Knowledge Source Selection In many practical settings, multiple knowledge sources may be available, denoted by

$$\mathcal{S} = \{\phi, s_1, \ldots, s_N\}, \tag{2}$$

where ϕ is a null source that always returns an empty string. A knowledge source selector P chooses the most appropriate source for a given query, i.e., $P(q, \mathcal{S}) \in \mathcal{S}$. Combined with the downstream inference, the full pipeline becomes

$$M\big(q, P(q, \mathcal{S})(q)\big). \tag{3}$$

Selective retrieval corresponds to the binary case $\mathcal{S} = \{\phi, s\}$, where the selector decides whether to invoke the retriever s or to skip retrieval via ϕ.

3.2 Self-Routing RAG: Overview

We propose Self-Routing RAG (SR-RAG), which generalizes selective retrieval by treating the LLM itself as a first-class knowledge source. Concretely, SR-RAG routes between external sources $\{S_{e1}, ..., S_{em}\}$ (e.g., several external corpora) and an internal source S_i (parametric knowledge). If S_i

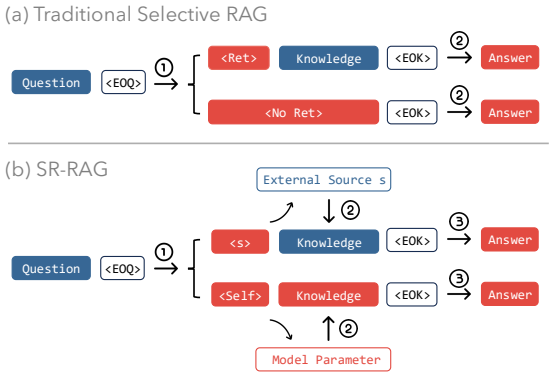


Figure 2: **SR-RAG expands selective retrieval into source routing.** Compared to traditional selective retrieval, SR-RAG allows the LLM to route among knowledge sources and to act as a knowledge source itself. Blue denotes external retrieved content and red denotes the LLM and its self-generated tokens.

is selected, the model does not directly answer. Instead, it first *verbalizes* query-relevant parametric knowledge into a compact context and then answers conditioned on this context. This makes the non-retrieval path an explicit knowledge consultation step rather than an absence of context.

Following prior work that streamlines adaptive RAG inference via special-token prediction (Asai et al., 2024; Wu et al., 2024a), SR-RAG uses three sets of special tokens to implement routing in a single left-to-right generation pass:

1. <EOQ>, which marks the end of the query and prompts source selection.[1]
2. A set of source tokens <s>, each representing a knowledge source $s \in \mathcal{S}$.[2]
3. <EOK>, which indicates the end of the knowledge context and triggers answer generation.

An illustration of the difference between SR-RAG and standard selective RAG is provided in Figure 2.

3.3 Self-Routing RAG: Training

We propose a fine-tuning recipe to improve any LLM's ability as the core model in the SR-RAG framework. Inspired by Wu et al. (2024a), we propose to mine self-supervision from existing question-answer data (q, a). Our pipeline uses the LLM itself as the internal source S_i along with the external knowledge source, requiring neither human routing labels nor a stronger teacher model. For simplicity, we formulate with $\mathcal{S} = \{S_i, S_e\}$, here. The more general form readily follows and is formulated in §B.

Data Construction Our central goals are (1) labeling which source is more helpful for a given instance (2) collecting the most helpful knowledge when parametric knowledge is preferred. To do so, our rollout process first collects n candidate contexts from each of the knowledge sources:

- *Parametric knowledge verbalization:* We use GenRead (Yu et al., 2023) to elicit n diverse verbalizations from the internal source S_i, denoted as $c_{i_1}, \ldots, c_{i_n}$.
- *External knowledge retrieval:* We retrieve n context passages from the external source S_e, denoted as $c_{e_1}, \ldots, c_{e_n}$.

Then, each candidate context c_j is scored by the log-likelihood of generating the gold answer:

$$l_j = \log p_M(a \mid q, c_j). \tag{4}$$

We then rank all $2n$ contexts by l_j and select the preferred source $s \in \{S_i, S_e\}$ as the source that appears most frequently among the top-ranked contexts. For convenience, we denote the highest- and lowest-scoring contexts from each source as $c_{i+}, c_{i-}, c_{e+}, c_{e-}$, respectively. The resulting training tuple is $(q, a, s, \{c_j, l_j\})$. The formal data collection algorithm is provided in §A.2.

Objective SR-RAG fine-tunes the LLM to perform (i) source selection, (ii) knowledge verbalization when S_i is selected, and (iii) answer generation conditioned on the selected knowledge. We use a two-stage training procedure.

Stage 1 We perform behavior cloning by jointly optimizing three losses:

1. $\mathcal{L}_{src}$, a cross-entropy loss that trains the model to predict the preferred source token immediately after <EOQ>:

$$\mathcal{L}_{src} = -\log p_M(\texttt{<s>} \mid q). \tag{5}$$

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

$$\mathcal{L}_{verb} = \begin{cases} -\log p_M(c_{i+} \mid q), & \text{if } s = S_i, \\ 0, & \text{if } s = S_e. \end{cases} \tag{6}$$

[1] This design aligns with Wu et al. (2024a) but differs from Asai et al. (2024). We find the additional <EOQ> helps the model allocate probability mass to source tokens without interfering with the query content.

[2] Our main setup uses a simple Wikipedia-based retrieval <Wiki> as the only S_e and <Self> for S_i. This formulation naturally extends to more than two sources by adding additional source tokens. We study a three-source setting in §B.

3. $\mathcal{L}_{ans}$, a cross-entropy loss on generating the answer conditioned on the preferred knowledge:

$$\mathcal{L}_{ans} = \begin{cases} -\log p_M(a \mid q, c_{i+}), & \text{if } s = S_i, \\ -\log p_M(a \mid q, c_{e+}), & \text{if } s = S_e. \end{cases} \tag{7}$$

We combine these objectives as

$$\mathcal{L}_{stage1} = \mathcal{L}_{src} + \mathcal{L}_{verb} + \mathcal{L}_{ans}. \tag{8}$$

Stage 2 To further improve the knowledge verbalization quality, we apply Direct Preference Optimization (DPO, Rafailov et al. (2023)) using self-generated preference pairs (c_{i+}, c_{i-}):

$$\mathcal{L}_{stage2} = \mathcal{L}_{src} + \mathcal{L}^{DPO}_{verb} + \mathcal{L}_{ans}, \tag{9}$$

$$\mathcal{L}^{DPO}_{verb} = \begin{cases} -\log \sigma\Big(\beta \log \frac{p_M(c_{i+}|q)}{p_{ref}(c_{i+}|q)} - \beta \log \frac{p_M(c_{i-}|q)}{p_{ref}(c_{i-}|q)}\Big), & \text{if } s = S_i, \\ 0, & \text{if } s = S_e. \end{cases} \tag{10}$$

Here M and ref are initialized from the stage 1 model checkpoint, and only M is updated. Conceptually, stage 2 distills a compute-intensive "system 2" process that searches over multiple verbalizations into a fast "system 1" behavior that produces a single high-quality knowledge verbalization during standard left-to-right decoding.

3.4 Self-Routing RAG: Inference

As shown in Figures 1 and 2, SR-RAG inference proceeds in a single left-to-right generation pass with three steps: source selection, knowledge collection, and answer generation.

Nearest Neighbor-Enhanced Source Selection A common way to choose a source is to threshold the model probability of a source token, such as $p_M(\texttt{<Wiki>} \mid q)$ (Asai et al., 2024; Wu et al., 2024a). In practice, only relying on model's uncertainty can be brittle and does not take into account model ability changes after the fine-tuning. To make source selection more robust and controllable, we augment token likelihoods with a dynamic kNN policy signal derived from additional *rollouts*. Concretely, we evaluate the fine-tuned model on a set of question-answer pairs[3]. For each pair, we compare the likelihood of generating the gold answer under different sources to decide a preferred source label. We then construct a *policy datastore* that maps each query to its preferred source, using the hidden representation at <EOQ> as the key. At test time, we retrieve the k nearest neighbors of the query in this representation space and form a neighbor-induced distribution over sources $p_D(\texttt{<s>} \mid q)$ from label counts. Finally, we select the source by applying a threshold on the combined score:

$$p_M(\texttt{<s>} \mid q) \times p_D(\texttt{<s>} \mid q). \tag{11}$$

This combination adapts the decision boundary to the fine-tuned model's behavior while preserving the efficiency of a single decoding pass. It also improves SR-RAG's interpretability and controlability as the datastore consists of explicit source assignments, which can be audited, modified, or expanded by practitioners to steer routing behavior.

Knowledge Collection and Answer Generation After selecting a source, SR-RAG gathers knowledge and generates the final answer within the same left-to-right pass. If the internal source S_i is selected, the model uses greedy decoding to verbalize a single compact knowledge context, which serves as a cost-efficient approximation to multi-sample verbalization. If an external source S_e is selected, decoding is paused to retrieve passages from the chosen index, which are appended as the knowledge context. In both cases, the system appends <EOK> and the model generates the final answer conditioned on the query and the knowledge.

4 Experimental Setup

4.1 Implementation Details of SR-RAG

Data Construction The main experiments are performed on two knowledge sources[4]: the 2018 English Wikipedia (<Wiki>) as the external knowledge source, and the LLM itself (<Self>) as the internal knowledge source. We use the official Wikipedia embeddings released by Karpukhin et al. (2020) and retrieve at a granularity of 100-word chunks. GenRead (Yu et al., 2023) is used to verbalize diverse knowledge contexts. GenRead clusters zero-shot knowledge verbalizations in as in-context demonstrations to verbalize diverse knowledge. We limit the verbalized knowledge chunks to a maximum of 150 tokens. From each knowledge source, we collect $n = 5$ knowledge contexts.

Training We fine-tune on a mixture of six short- and long-form knowledge-intensive datasets: Wizard of Wikipedia (Dinan et al., 2019), Natural

[3]Empirically, we reuse the training set so that no additional data or supervision is required.

[4]We present multi-source extensions of SR-RAG in §B.

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 instruction tuning data shown effective in Asai et al. (2024). After running the data construction algorithm, 46.9% of the instances are labeled with <Self> and the rest are labeled with <Wiki> as the preferred knowledge source, averaged across three LLMs. Full training details are provided in §A.2.

Inference To construct the policy datastore, we use the representation from a middle layer in the fine-tuned LLM[5], as middle layers are found to be indicative of LLM truthfulness (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 and construct $p_D(\texttt{<s>}|q)$ from the counts of each knowledge source as the preferred source. Then, we impose a threshold τ on $p_M(\texttt{<Wiki>}|q) \times p_D(\texttt{<Wiki>}|q)$ to decide whether the external knowledge source should be selected.[6]

4.2 Evaluation

Datasets and Metrics We evaluate on a diverse set of four knowledge-intensive NLP tasks. **PopQA** (Mallen et al., 2023) is a free-form long-tail open-domain question answering dataset. Following Asai et al. (2024), we use the subset of 1,399 questions that aims to test knowledge of long-tail entities. **TriviaQA** (Joshi et al., 2017) is an established open-domain question answering dataset that features relatively complex and diverse questions. We use the same test split and retrieval setup as in Asai et al. (2024). **PubHealth** (Zhang et al., 2023) is a fact-checking dataset focusing on the public health domain. Finally, **ARC Challenge** (Bhakthavatsalam et al., 2021) is a multiple-choice question answering dataset featuring grade-school level science questions. Following common practice, we perform lexical post-processing 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 baselines that cover various training and inference strategies. (1) First, using the LLM before fine-tuning, we compare with either always retrieving or always verbalizing with GenRead. (2) As illustrated in Figure 2, our main baseline is the state-of-the-art prior *selective retrieval* pipeline, combining the ideas from He et al. (2021), Asai et al. (2024), and Wu et al. (2024a). 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, we apply a uniform threshold of 0.2 to the likelihood of the retrieval token for selective retrieval. (3) Always retrieving with the fine-tuned LLM after two-staged SR-RAG fine-tuning.

5 Results

5.1 Knowledge Verbalization Alters Knowledge Source Preference

To motivate SR-RAG, we first show that knowledge verbalization can substantially change when an LLM benefits from retrieval. In a pilot study on Llama-2-7B-Chat using four datasets from our training mixture, we compare the likelihood of generating the gold answer under three conditions: no context (blue), the most helpful GenRead verbalization (c_{i+}, orange), and the most helpful retrieved passage (c_{e+}, red). As shown in Figure 3, GenRead reverses the preferred knowledge source for a large fraction of instances, including 16% of Natural Questions and over 30% on the other datasets. This indicates that selective retrieval methods that omit verbalization can underestimate the LLM's effective capability without retrieval, leading to suboptimal source preference labels and weaker performance when retrieval is skipped. Additional details of the study setup are provided in §A.2.

5.2 SR-RAG Improves Accuracy with Reduced Retrieval Frequency

Table 1 shows the end-to-end generation performance on three LLMs, demonstrating the advantage of SR-RAG over always retrieving, selective retrieval, and other baselines. Relative to always retrieving with the pretrained model, SR-RAG improves accuracy by 19.3% (Llama-2-7B-Chat), 13.4% (Phi-3.5-Mini-Instruct), and 5.9% (Qwen2.5-7B-Instruct). Compared to the selective retrieval baseline, SR-RAG achieves 8.5%/2.1%/4.7% higher accuracy while triggering 26%/40%/21% fewer retrievals across the three

[5]Layer 15 for Llama-2-7B-Chat and Phi-3.5–Mini-Instruct and layer 11 for Qwen2.5-7B-Instruct. We provide further visualizations and discussions of layer selection in §C.2.

[6]$\tau = 0.1$ for Llama-2-7B-Chat and $\tau = 0.2$ for other models.

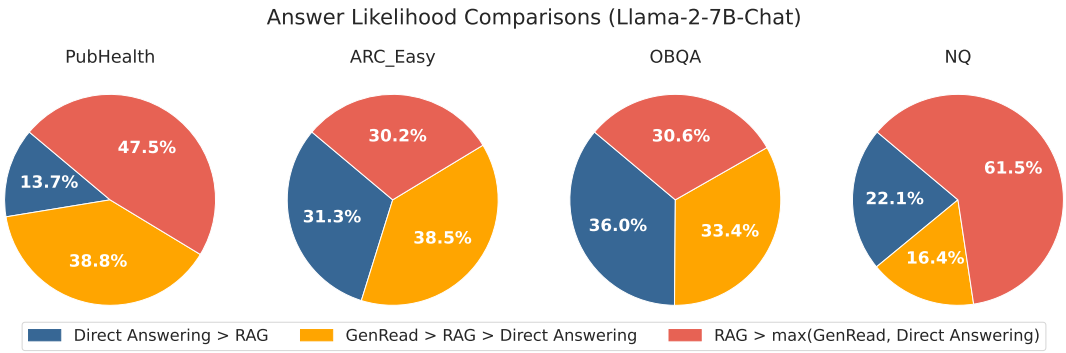


Figure 3: **Knowledge verbalization significantly affects the LLM ability boundary.** For a large number of instances (16.4% - 38.8%, orange), GenRead reverses the knowledge source preferences: without considering GenRead, RAG dominates over parametric knowledge.

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%	**0.730**	40%	**0.630**	29%	**0.648**	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%
Qwen2.5-7B-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. Across three backbone LLMs, SR-RAG consistently outperforms selective RAG and always retrieving while using a much lower retrieval budget.

Method	PopQA	TriviaQA	PubHealth	ARC	Average
Accuracy (Verbalization $\geq$ 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**. Llama-2-7B-Chat is used. SR-RAG achieves the best averaged performance in both evaluation settings.

models, respectively. Notably, the selective retrieval baseline saves only 8–15% retrievals and achieves nearly identical accuracy to always retrieving, suggesting that skipping retrieval alone is not sufficient to reliably identify queries that can be handled without external context. In contrast, SR-RAG reduces retrieval by 20–40% while maintaining or improving accuracy, driven by accurate source selection and a stronger non-retrieval path.

Remarkably, without threshold tuning, SR-RAG can *dynamically adapt* its retrieval behavior. For

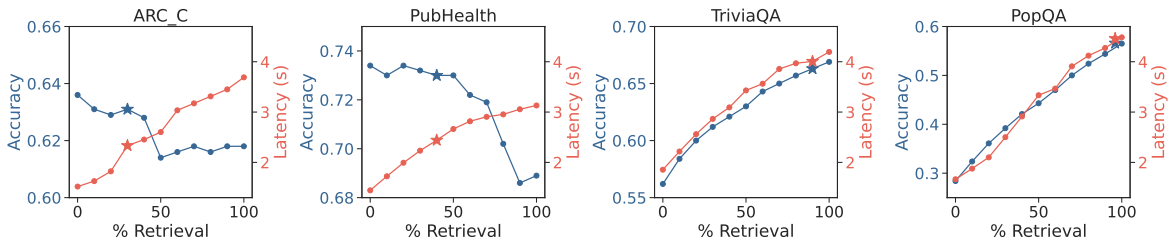


Figure 4: **Accuracy and system latency of SR-RAG.** We show the performance of fine-tuned Llama-2-7B-Chat with different verbalization frequencies. SR-RAG's source selection policy (marked with stars) achieves near-optimal accuracy-efficiency trade-off without dataset-specific thresholds.

Training	Inference	PopQA		TriviaQA		PubHealth		ARC		Average	
		ACC	%RAG	ACC	%RAG	ACC	%RAG	ACC	%RAG	ACC	%RAG
SR-RAG	SR-RAG	0.566	96%	**0.664**	89%	**0.730**	40%	**0.630**	29%	**0.648**	64%
SR-RAG	SR-RAG w/o. kNN	0.558	94%	0.658	77%	0.694	72%	0.627	56%	0.634	75%
SR-RAG w/o. kv. label	SR-RAG	**0.568**	100%	0.644	100%	0.598	100%	0.629	84%	0.610	96%
SR-RAG w/o. $\mathcal{L}_{verb}^{DPO}$	SR-RAG	0.564	98%	0.645	100%	0.674	100%	0.581	66%	0.616	86%

Table 3: **Ablation studies on SR-RAG.** Llama-2-7B-Chat is used as the LLM.

PopQA, which emphasizes long-tail facts, SR-RAG retrieves external knowledge more frequently. For PubHealth and ARC, where the model can often rely on internal knowledge, SR-RAG skips retrieval more and can outperform always retrieving. We find that this flexibility also extends to selecting a specialized external source when multiple are present (§B). We also present qualitative examples in the appendix (Figures 7 and 8).

5.3 SR-RAG Makes Accurate Source Selection Decisions

Can SR-RAG choose the appropriate knowledge source? Table 2 compares SR-RAG with Self-RAG (Asai et al., 2024) and an SR-RAG variant without kNN augmentation. We evaluate decision quality under two criteria: (1) abstaining from retrieval when skipping retrieval does not harm performance and (2) retrieving only when retrieval yields strictly better context. Under the first criterion, all methods perform well, and SR-RAG achieves the best overall accuracy. Under the stricter second criterion, SR-RAG substantially improves AUROC, outperforming Self-RAG by 14.7% on average. Removing kNN augmentation leads to large drops in both accuracy and AUROC, highlighting the benefit of adapting source selection to fine-tuning-induced shifts in model behavior. We visualize the hidden state space in §C.2 to provide further intuitions.

5.4 System Efficiency

We measure end-to-end latency using Llama-2-7B-Chat under a batched inference setting (details in §A.3). Figure 4 shows the accuracy-latency trade-off as we vary retrieval frequency. As expected, latency improves as fewer instances invoke retrieval, and the optimal retrieval rate varies across datasets. Importantly, SR-RAG's learned policy achieves a near-optimal accuracy-efficiency trade-off across datasets without dataset-specific threshold tuning. These results indicate that SR-RAG improves both accuracy and efficiency in a robust manner.

5.5 Ablation Study

We ablate SR-RAG components on Llama-2-7B-Chat in Table 3. Disabling kNN-based routing (w/o. kNN) increases retrieval frequency and reduces accuracy, confirming that kNN augmentation improves robustness to ability shifts introduced by fine-tuning. Removing knowledge verbalization during preference labeling (w/o. kv. label) causes the model to over-rely on retrieval and degrades performance, showing that verbalization-aware supervision is important for accurate routing. Finally, ablating stage-2 preference alignment (training stage 2 with the stage-1 objective) reduces verbalization quality and leads to substantially more retrievals, consistent with weaker internal knowledge generation. Together, these results indicate that all three components contribute materially to SR-RAG's overall gains. We provide additional

analyses in §C including the choice for source labeling heuristics, the size of the policy datastore, and other hyperparameters.

6 Conclusion

We present SR-RAG, a novel retrieval-augmented generation (RAG) framework that tightly integrates selective retrieval with knowledge verbalization. By reformulating selective retrieval as a knowledge source selection problem, SR-RAG enables the LLM to not only choose between external and internal knowledge sources but also to serve as a knowledge source itself. During inference, SR-RAG leverages internal hidden states and a nearest-neighbor policy to make accurate, adaptive source selection decisions. Extensive experiments show that SR-RAG significantly improves answer accuracy while reducing retrieval frequency and latency, offering a scalable and reliable path forward for more efficient, knowledge-aware RAG systems.

Limitations

Our study has several limitations. First, we focus on 7B-class backbones, so routing behavior and absolute accuracy may shift with larger or distilled models. We expect the qualitative trends to remain and view scaling studies as a direct next step. Second, SR-RAG adds system complexity: training-time supervision requires generating and scoring multiple candidate contexts, and inference uses a kNN policy datastore with additional engineering and memory overhead compared to likelihood-only thresholding. In our implementation, kNN search is lightweight when cached on GPU, and performance is not highly sensitive to datastore size, suggesting that compact deployments are feasible (Figure 6). Third, our source labeling uses answer log-likelihood as a proxy for context utility, which aligns with our objective but may not fully capture notions such as completeness or faithfulness. Nevertheless, this proxy yields consistent end-to-end gains across benchmarks, and alternative supervision signals (e.g., verifier- or reward-based) can be integrated into the same routing formulation. Finally, while SR-RAG achieves favorable accuracy-latency trade-offs without dataset-specific threshold tuning (Figure 4), it is not always locally optimal on every dataset. We partially address this with kNN augmentation and robustness analyses, and improving routing optimality under distribution shift remains an important direction.

Ethics Statements

In this section, we describe the ethical considerations related to this paper.

Potential Risks SR-RAG could introduce or amplify risks beyond standard RAG. First, routing choices may systematically favor sources that reflect population or topical skews, resulting in uneven performance across demographics or domains. Optimizing for latency or recency can also shift answers toward lower-quality or sensational sources (source-selection bias). Second, because SR-RAG may consult private and public corpora, mis-routing can increase the chance of privacy leakage (e.g., retrieving internal snippets when public evidence suffices) or enable membership inference about whether certain private documents exist. To mitigate these risks, we recommend that users and deployers: (1) adopt a "least-privilege" routing policy by default and require explicit consent before accessing private indices, (2) implement auditing, retention limits, and user-initiated deletion to reduce privacy and compliance risks.

Artifact Creation and Usage We release only code and configuration necessary to reproduce results, plus model-agnostic routing checkpoints and labels that do not include third-party text, and we do not redistribute original corpora. We plan to release the code and checkpoints with MIT License. Self-RAG mixture (Hugging Face `selfrag/selfrag_train_data`) is under the MIT License, and we use it for research without redistributing source documents; Self-RAG checkpoints are in a GitHub repository under the MIT License, and the Llama 2 base is governed by Meta's Llama 2 Community License, and we use them for non-commercial academic research with required notices; TriviaQA is under Apache-2.0, and we use it for evaluation with attribution; ARC-Challenge on Hugging Face is under CC BY-SA 4.0, and we use it for evaluation with attribution; PubHealth is under the MIT License, and we use it for evaluation only; PopQA is under the MIT License, and we use it for evaluation only without redistribution. We use the datasets as-is and rely on the original distributions to ensure no question may cause privacy or ethics concerns.

AI Assistant Use AI assistants, specifically ChatGPT, are used only for improving the paper writing and the appearance of the figures.

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.

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.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. 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.

Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. 2025. Adacot: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning. *arXiv preprint arXiv:2505.11896*.

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. 2025a. Adaptive retrieval without self-knowledge? bringing uncertainty back home. *CoRR*, abs/2501.12835.

Viktor Moskvoretskii, Maria Lysyuk, Mikhail Salnikov, Nikolay Ivanov, Sergey Pletenev, Daria Galimzianova, Nikita Krayko, Vasily Konovalov, Irina Nikishina, and Alexander Panchenko. 2025b. Adaptive retrieval without self-knowledge? bringing uncertainty back home. *CoRR*, abs/2501.12835.

National Center for Biotechnology Information (NCBI). Pubmed. `https://pubmed.ncbi.nlm.nih.gov/`. U.S. National Library of Medicine (NLM), National Institutes of Health (NIH).

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. 2024a. 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.

Di Wu, Jia-Chen Gu, Fan Yin, Nanyun Peng, and Kai-Wei Chang. 2024b. Synchronous faithfulness monitoring for trustworthy retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9390–9406, Miami, Florida, USA. Association for Computational Linguistics.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6233–6251, Bangkok, Thailand. Association for Computational Linguistics.

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.

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 SR-RAG: Further Details

A.1 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 the LLM itself as knowledge source.
k	number of neighbors retrieved for source policy inference

Table 4: **A summary of the key symbols.**

A.2 Training Details

Dataset Construction Algorithm Algorithm 1 presents the full algorithm for constructing training data and labeling knowledge source preferences. GenRead is executed independently on each training data subset. We adopt instance-level notation for clarity. The pipeline naturally scales to additional knowledge sources by applying knowledge collection and likelihood evaluation in parallel across sources.

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

GenRead Prompt We implement GenRead following the setup in Yu et al. (2023). In the second verbalization round, five clusters with five in-context examples each are used. All datasets except ASQA follow the general prompt shown in Figure 5. For ASQA, we include an additional instruction to handle ambiguity: "`If the question is ambiguous, generate multiple documents for each possibility.`"

```
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 Table 5 summarizes the training and validation splits, including the proportion of examples where self-verbalized knowledge is preferred over retrieved knowledge.

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 the training and validation data with verbalization percentages.** Llama = Llama-2-7B-Chat, Phi = Phi-3.5-Mini-Instruct, and Qwen = Qwen2.5-7B-Instruct.

Training Process To fully leverage the backbone LLM's ability to follow natural language instructions, both SR-RAG fine-tuning and inference use the following prompt that interleaves special tokens with natural language:

```
Question: {question} Background
knowledge: <EOQ>
<s> {knowledge} <EOK> Answer: {answer}
```

During training, the loss is computed on the knowledge part only if <s> is <Self>. If <s> is <Wiki>, we augment c_{e+} by randomly appending $\max(p - 1, 0)$ retrieved contexts, where p is sampled from a Poisson distribution with $\lambda = 2$. This data augmentation improves the LLM's robustness to different retrieval strategies and various levels of retrieval

quality. 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, β = 0.3 for DPO, and train for another epoch. All the experiments are performed on a machine with eight A800 (80GB) GPUs and a machine with eight A6000 GPUs. On eight A800 (80GB) GPUs, the two-stage training takes approximately 10 hours for a 7B-sized model.

A.3 Latency Formulation

To evaluate the inference efficiency of SR-RAG, we measure the latency in a realistic batched inference setup, where the system handles a batch of B = 10 queries and returns the results for all queries. For the latency experiments presented in the paper, we decompose the system latency as follows:

- **Source Selection Time** (T_d): The time for the knowledge source selector to determine whether to retrieve from external sources or rely on parametric knowledge.
- **Retrieval Latency** (T_{rs}): The time for fetching external knowledge from the database if the model chooses to retrieval from an external knowledge source s. In our batched setting, we calculate T_{rs} by performing a batched retrieval for all the B instances that require retrieval and report the per-item latency.
- **Verbalization Latency** (T_v): The time for the LLM to verbalize parametric knowledge if the LLM itself is selected as the knowledge source.
- **Generation Latency** (T_g): The time for the LLM to generate the response, conditioned on either retrieved or verbalized knowledge.

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

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

We have the following remarks:

- This formulation assumes that both the retrieval index and the source selection datastore are pre-constructed. This assumption is reasonable as these indices are only constructed once and then constantly reused.
- We choose the batched setup due to the complexity of the retrieval system. For instance, in our implementation of Wikipedia search, it takes around five to ten seconds per instance to encode the query and retrieve the most relevant context chunks. The batched setting amortizes the latency of single retrieval.
- As the nearest-neighbor search only involves one matrix product for similarity calculation and one top-k operation, T_d is generally very small. In fact, we find T_d (0.01s) $<<$ T_g (0.1s) $<$ T_v (1s). On the other hand, T_{rs} is the major bottleneck of the pipeline. As a result, in an online setting, the system's efficiency gain directly converges to the percentage of retrievals it is able to avoid.

B SR-RAG: Multi-Source Extensions

Beyond routing between the LLM itself and an external knowledge source, SR-RAG's formulation is naturally compatible with routing between multiple external knowledge sources as well. In this section, we provide further experiments on routing between three knowledge sources and discuss other potential multi-source applications.

B.1 Experiment: Routing between General and Specialized Sources

We perform experiments on the PubMed corpus (National Center for Biotechnology Information (NCBI)), which contains professional biomedical literature. Compared to Wikipedia and the LLM itself, PubMed has the unique advantage of featuring long-tail specialized biomedical knowledge. Meanwhile, it falls short at providing knowledge for many general factuality questions, posing a challenge to accurately routing the knowledge source.

Based on this new setup, we formulate SR-RAG with three knowledge source tokens: `<Self>`, `<Wiki>`, and `<Pubmed>`. We leverage the MedRAG toolkit (Xiong et al., 2024) to implement retrieval on PubMed. The training and inference strategies of the two-source SR-RAG recipe naturally extends to this formulation and are followed in our experiments here. For training and evaluation, we additionally use PubMedQA (Jin et al., 2019), a widely used biomedical question answering dataset. PubMedQA features short questions extracted from PubMed abstracts and provides associated ground truth yes/no/maybe answers. We sample 10k artificially generated documents for training (combined

Training	Inference	PubMedQA	PopQA	TriviaQA	PubHealth	ARC	Average
			`Llama-2-7B-Chat`				
No Fine-tuning	GenRead	0.544	0.247	0.616	0.515	0.605	0.505
	Always RAG, Wiki	0.636	0.529	0.641	0.457	0.546	0.562
	Always RAG, PubMed	0.656	0.157	0.470	0.612	0.537	0.486
SR-RAG (3-source)	Always RAG, Wiki	0.342	**0.587**	0.639	0.668	0.639	0.572
	Always RAG, PubMed	0.672	0.325	0.539	0.713	0.645	0.642
	Always Verbalizing	0.388	0.305	0.572	**0.724**	0.631	0.579
	SR-RAG	**0.680**	0.583	**0.647**	**0.724**	**0.655**	**0.677**
	(% <Wiki>)	6%	95%	88%	17%	4%	29%
	(% <Pubmed>)	91%	4%	7%	70%	86%	64%
	(% <Self>)	3%	1%	5%	13%	10%	8%
			`Phi-3.5-Mini-Instruct`				
No Fine-tuning	GenRead	0.376	0.331	0.567	0.442	**0.840**	0.511
	Always RAG, Wiki	0.460	0.541	0.594	0.549	0.771	0.583
	Always RAG, PubMed	0.676	0.143	0.356	0.447	0.726	0.470
SR-RAG (3-source)	Always RAG, Wiki	0.486	**0.580**	0.634	0.676	0.805	0.636
	Always RAG, PubMed	0.738	0.302	0.494	0.644	0.813	0.598
	Always Verbalizing	0.498	0.327	0.510	0.704	0.819	0.572
	SR-RAG	**0.736**	0.560	**0.639**	**0.716**	0.824	**0.695**
	(% <Wiki>)	1%	95%	87%	54%	0%	47%
	(% <Pubmed>)	97%	3%	10%	18%	27%	31%
	(% <Self>)	2%	2%	3%	28%	73%	22%
			`Qwen2.5-7B-Instruct`				
No Fine-tuning	GenRead	0.418	0.334	0.626	0.676	0.875	0.586
	Always RAG, Wiki	0.612	0.563	**0.667**	0.446	0.916	0.641
	Always RAG, PubMed	0.698	0.129	0.469	0.401	0.821	0.504
SR-RAG (3-source)	Always RAG, Wiki	0.358	**0.576**	0.639	0.637	0.818	0.606
	Always RAG, PubMed	0.726	0.307	0.503	0.676	0.824	0.607
	Always Verbalizing	0.522	0.303	0.550	**0.695**	0.828	0.580
	SR-RAG	**0.730**	0.572	0.651	0.681	**0.834**	**0.694**
	(% <Wiki>)	15%	96%	86%	20%	0%	43%
	(% <Pubmed>)	79%	1%	13%	48%	39%	36%
	(% <Self>)	6%	3%	1%	32%	61%	21%

Table 6: **Three-source results.** We route between retrieving from **Wikipedia**, retrieving from **PubMed**, or **self-verbalizing** knowledge. For each model we report task accuracy and the selection frequencies of the individual sources `wiki/PubMed/self` under SR-RAG's LLM+kNN knowledge source routing scheme.

with the data mixture used in the main text) and 500 examples from the expert-labeled subset for evaluation. The same set of LLMs from three families (Llama, Qwen, Phi) are trained and evaluated.

In Table 6, we present the three-way routing results of SR-RAG and compare with always-RAG as well as GenRead with the model before fine-tuning. Our main findings are as follows:

- **Adding PubMed as an additional knowledge source does not harm its effectiveness.** The results on datasets other than PubMedQA align with the performance of two-source SR-RAG reported in the main text.
- **SR-RAG training improves the model's ability to verbalize its knowledge** on PubMedQA as well while being much more efficient than GenRead, as shown by the better performance of SR-RAG + Always Verbalizing compared to GenRead.
- **SR-RAG intelligently routes the best knowledge source for different workloads.** For

PubMedQA, PubHealth, and ARC, where PubMed can outperform Wiki by a large margin, SR-RAG frequently self-selects PubMed as the knowledge source. On the other hand, on general factuality benchmarks such as PopQA and TriviaQA, SR-RAG selects Wikipedia more frequently. More importantly, SR-RAG achieves this behavior end-to-end without the need for any manual interventions.

- **On PubMedQA, the end-to-end SR-RAG achieves the best performance compared to the baselines.** The trend is consistent across three different LLMs.

Overall, these experiments confirm the effectiveness of extending SR-RAG to three sources and suggests a promising outlook for further extension.

B.2 Discussion: Further Applications

SR-RAG has numerous useful real-world applications and exhausitively experimenting on them is beyond the scope of this paper's experiments. In this section, we discuss several potential applications that SR-RAG naturally supports.

Cost-Aware Routing Querying different knowledge sources brings distinct latency costs. SR-RAG's preference label can be constructed such that the cheapest useful knowledge source is preferred, improving the latency while preventing performance degradation. In addition, varied levels of compute are required to verbalize the required knowledge for different types of queries. SR-RAG can be used to model this pattern by formulating multiple internal knowledge sources associated with different levels of compute. This philosophy connects with the literature on adaptive reasoning budget (Lou et al., 2025).

Recency-Aware Routing Many user queries have an inherent temporal dimension (e.g., "today," "as of Sept. 2025," fast-moving entities) where stale knowledge degrades correctness. SR-RAG can expose "freshness-tiered" sources and learn a preference label that prioritizes the most up-to-date tier when temporal cues or out-of-vocabulary entities are detected. The router can condition on time expressions, named-entity recency features, and source timestamps. When the model's self-estimate is uncertain or predates the query's time scope, it escalates to a recency-guaranteed index.

Privacy-Aware Routing Queries often mix public facts with sensitive tenant data (e.g., customer tickets, internal codenames, emails). SR-RAG can represent privacy tiers as distinct sources and train preference labels under a "least-privilege first" policy: prefer private indices when identifiers or confidentiality cues are present and fall back to public only when the query is non-sensitive and private stores are irrelevant.

C Further Analyses

C.1 Self-Routing RAG: Hyperparameters

In this section, we analyze other hyperparameters in SR-RAG.

Source Labeling Heuristics In SR-RAG, the default design for source preference labeling is to collect the knowledge from each source and to select the top-ranked source among the top 50% of knowledge contexts in terms of contribution to the likelihood of the answer. In this section, we compare this strategy to a number of alternative heuristics:

- **Best Single Likelihood:** Selecting the source that produces the knowledge leading to the highest answer likelihood.
- **Best Average Likelihood:** Selecting the source that leads to the highest answer likelihood, averaged over all knowledge contexts.
- **Best All Rank:** Selecting top-ranked knowledge source using all the knowledge context instead of top-50%.

In Table 7, we present the F1 score and AUROC using Llama-2-7B-Chat as the LLM. Correctness is defined as preferring self-verbalized knowledge when it is better than or equal to performing retrieval in terms of the downstream question answering performance. Compared to three baseline methods, the heuristics SR-RAG uses achieves the best F1 and AUROC over the short-form and closed-set subset of the training set. Building upon this paper's results, future work can further study leveraging more advanced uncertainty quantification methods (Wu et al., 2024b; Moskvoretskii et al., 2025b) to build more accurate source selection labels.

Datastore Size To study the influence of the kNN policy datastore size on the performance of SR-RAG, we randomly sample a subset of the training

Method	ARC_Easy		NQ		OBQA		Fever		Average	
	F1	AUROC	F1	AUROC	F1	AUROC	F1	AUROC	F1	AUROC
Best Single Likelihood	0.640	0.572	0.320	0.551	0.660	0.606	0.560	0.650	0.545	0.585
Best Average Likelihood	0.730	**0.692**	0.420	0.626	**0.720**	**0.643**	0.630	0.703	0.630	0.666
Best All Rank	0.700	0.691	**0.470**	0.632	0.700	0.621	0.630	0.674	0.625	0.655
SR-RAG	**0.740**	0.691	0.440	**0.635**	**0.720**	0.635	**0.670**	**0.711**	**0.643**	**0.668**

Table 7: **Source labeling accuracy of different heuristics.** F1 and AUROC scores for different methods across datasets. Llama-2-7B-Chat is used as the LLM. The best score per column is boldfaced.

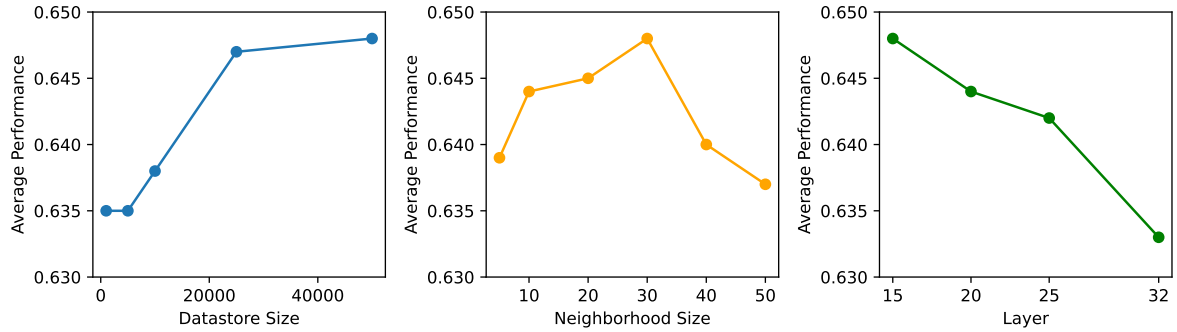


Figure 6: **Hyperparameter settings for the kNN policy datastore with Llama-2-7B-Chat.**

set (50k samples in total) to construct the datastore. As shown in Figure 6 (blue), reducing the kNN policy datastore to 25k (half) has minimal impact. Even with just 1k examples, performance drops only slightly, suggesting potential for lightweight and memory-efficient deployments.

Number of Neighbors The number of neighbors retrieved from the datastore determines the localness of the focus in the hidden representation space. While task-specific neighborhood size can be tuned, SR-RAG generally works well with 10 to 30 neighboring data points (Figure 6 middle), indicating its robustness. However, using a too small (e.g., 5) or a too large (e.g., more than 50) neighborhood harms the performance of SR-RAG.

Layer for Hidden State We further study the impact of layer selection on SR-RAG's performance in Figure 6 part 3 (green). Overall, we observe that using a middle layer's hidden states has better performance over using the last layer's. In the next section, we visualize the hidden state space across LLMs and show that middle layers tend to learn better representations for the model uncertainty as well as task types.

C.2 Hidden State Space of the Source Reflection Token

Does SR-RAG training allow the model to condense its knowledge of tasks, model uncertainty, and source characteristics into the representation of <EOQ>?

Uncertainty Encoding in <EOQ> In Figure 9, we visualize <EOQ> hidden states on PopQA using t-SNE. Clear separation emerges between instances where self-verbalized knowledge is helpful (green) and unhelpful (red). Middle layers show stronger clustering than final layers, supporting the use of these representations for kNN-based source selection.

Implicit Task Clustering We further randomly sample 500 data points from the SR-RAG train set and visualize their hidden states. Figure 10 shows that SR-RAG also learns to cluster instances by task type (e.g., fact-checking, closed QA, long-form generation). This suggests that source reflection tokens carry semantically meaningful and task-sensitive representations on top of uncertainty information. These behaviors serve as the foundation of our nearest neighbor-based source selection approach.

Source for c_{i+} and c_{i-}	PopQA	TriviaQA	PubHealth	ARC	Average
%Verb $\geq$ RAG (↑)					
Self-Generated Knowledge	**33.0%**	**56.6%**	**73.4%**	**63.5%**	**56.6%**
Distilled Knowledge	28.2%	55.3%	69.7%	62.3%	53.9%
Verbalization Performance (↑)					
Self-Generated Knowledge	**0.302**	**0.572**	**0.734**	**0.634**	**0.561**
Distilled Knowledge	0.282	0.553	0.697	0.606	0.535

Table 8: **Effect of knowledge distillation.** Comparison between training with self-generated knowledge versus knowledge distillation for the verbalization branch in SR-RAG. Llama-2-7B-Chat is fine-tuned either with self-verbalized knowledge or the knowledge verbalized from Llama-3.3-70B-Instruct.

C.3 Can SR-RAG learn more from self-generated knowledge?

In SR-RAG training, self-verbalized knowledge contexts from the LLM are utilized to fine-tune the model via behavior cloning (stage 1) or preference alignment (stage 2). Alternatively, is it possible that the model can learn to verbalize even higher-quality knowledge from off-policy samples generated by a stronger LLM? We investigate this hypothesis by running GenRead-based knowledge verbalization on Llama-3.3-70B-Instruct and directly use the knowledge to prepare SR-RAG training data with Llama-2-7B-Chat. Table 8 compares the performance of fine-tuning Llama-2-7B-Chat with this data versus using self-generated knowledge for SR-RAG. Interestingly, using self-generated positive and negative knowledge pairs is more beneficial for unlocking the model's ability to generate useful knowledge, outperforming the alternative by 4.9% higher answer accuracy and 5.0% higher win rate over retrieval. It is also possible that the ability of the teacher model needs to be close to the fine-tuned model for its knowledge to be useful. We leave this investigation to future work.

C.4 Discussion: Efficiency and Applicability

In this section, we briefly discuss the efficiency and applicability of SR-RAG.

Training Efficiency SR-RAG introduces additional offline computation because source labels are constructed by collecting multiple candidate contexts per source (retrieved passages and self-verbalizations) and scoring them with answer likelihood (§A.2). We view this cost as a one-time post-training expense that can be amortized across deployments, and it requires no extra human annotation or stronger teacher models. In addition, our two-stage recipe mirrors common modern post-training pipelines that combine supervised fine-tuning with preference alignment, and it can be incorporated without substantial changes to standard training infrastructure.

Inference Efficiency SR-RAG preserves efficient inference by executing routing, knowledge collection, and answer generation in a single left-to-right pass (Figure 1). When the internal source is selected, the model produces one compact knowledge verbalization instead of running multi-sample elicitation; when an external source is selected, decoding pauses only to issue retrieval, and overall latency is largely governed by retrieval frequency (Figure 4 and §A.3). The kNN policy adds a lightweight computation when the datastore is cached on GPU, and our hyperparameter analyses show that routing quality is not highly sensitive to datastore size, suggesting that compact or compressed datastores can be effective (Figure 6).

Applicability SR-RAG is designed for practical routing settings where systems must choose among heterogeneous sources. First, SR-RAG achieves favorable accuracy-efficiency trade-offs without dataset-specific threshold tuning, which reduces the manual effort required to deploy selective retrieval across diverse workloads (Figure 4). Second, the policy datastore improves interpretability because it stores explicit source assignments that can be inspected and debugged, rather than relying solely on opaque likelihood thresholds. Third, the same datastore also enables controllability: practitioners can audit and modify routing behavior by editing, filtering, or augmenting stored examples (for instance, enforcing least-privilege routing or overriding problematic regions of the decision boundary). Finally, SR-RAG naturally extends beyond the two-source

setup, and our three-source experiments illustrate that the framework can route to specialized corpora (e.g., PubMed) when needed while retaining strong performance on general workloads (§B).

C.5 Qualitative Study

In Figure 7 and Figure 8, we show two qualitative examples from TriviaQA with the fine-tuned Llama-2-7B-Chat as the backbone LLM. For both examples, we show the LLM's prediction conditioned on both retrieved knowledge and the verbalized knowledge. While the verbalized knowledge is more concise than retrieved paragraphs, it better identifies the correct answers for both questions. This is because the core information (i.e., "first name", "island") only takes up a small portion of the query. As a result, dense retrieval is noise-prone and brings in only distantly relevant information. By contrast, the LLM as a knowledge generator dynamically adjusts its focus, guiding the knowledge generation to include the core information. As a result, the LLM can correctly answer both questions with self-verbalized knowledge but provides incorrect answers with the low-quality retrieved knowledge. In the source selection process, the LLM directly predicts the source preference $p_M(\texttt{<s>}|q)$. In addition, questions from the train set sharing the same topic and difficulty level are retrieved to form the source preference distribution $p_D(\texttt{<s>}|q)$. In the first example (Figure 7), both distributions prefer the LLM itself as the knowledge source. By contrast, in the second example (Figure 8), $p_M(\texttt{<s>}|q)$ has a high uncertainty and $p_D(\texttt{<s>}|q)$ corrects the source distribution to select the LLM itself as the knowledge source.

Question: In the book David Copperfield what was Mr. Micawber's first name?
Answer: Wilkins

Retrieved Knowledge: [1] David Copperfield (1999 film) a visit from Mr. Peggotty, back from Australia. He brings news that Emily has made a full recovery and that Mr. Micawber has established himself as a successful magistrate and bank manager. The story closes with the birth of David and Agnes's third child - a girl. Betsey Trotwood's wish finally comes true after nearly 30 years, as David decides that the baby will be christened Betsey Trotwood Copperfield, in honour of her godmother. [2]... (following knowledge chunks omitted)
Prediction (with retrieved knowledge): John Micawber, Sr., is a fictional character in Charles Dickens's 1850 novel David Copperfield. He is the father of David Copperfield's godfather, Mr. Micawber, Jr., and is portrayed as a kind and gentle man who is often in financial difficulties. Mr. Micawber's first name is Charles, not John, as is often mistaken.

Verbalized Knowledge: David Copperfield is a novel by Charles Dickens, published in 1849-50. The novel follows the life of David Copperfield, a young man who grows up in Victorian England. Mr. Micawber is a character in the novel, a kind and well-meaning man who is a friend of David's father. Mr. Micawber's first name is Wilkins.
Prediction (with verbalized knowledge): Wilkins Micawber. Mr. Micawber is a character in Charles Dickens 's novel David Copperfield. He is a kind and well - meaning friend of David 's father, Mr. Copperfield. Mr. Micawber is known for his optimistic outlook on life and his tendency to get into financial difficulties.

Neighbor Questions:
- What was Mr. Smith's first name in Mr. Smith goes to Washington? (<Self> > <Wiki>)
- Who was the original singer of Knocking on Heaven's Door? (<Self> > <Wiki>)
- When did sir Arthur Conan Doyle wrote his first book? (<Wiki> > <Self>)
... (27 questions omitted)
$p_D(\texttt{<s>}|q)$: P(<Wiki>) = 9/30 = 0.3, P(<Self>) = 21/30 = 0.7.
$p_M(\texttt{<s>}|q)$: P(<Wiki>) = 0.146, P(<Self>) = 0.854.
Final Source Decision: <Self>

Figure 7: **Qualitative Example 1.** Example of SR-RAG inference where $p_D(\texttt{<s>}|q)$ agrees with $p_M(\texttt{<s>}|q)$. As the model itself is selected as the knowledge source, the retrieval will be abstained. The retrieved knowledge and the corresponding prediction are shown only for illustration.

Question: From 1791 to 1803, Toussaint L'Ouverture led a revolt against France on which island?
Answer: Haiti

Retrieved Knowledge: [1] Saint-Domingue expedition slave revolt that led to the abolition of slavery in 1793 by the civil commissioners Sonthonax and Polverel, in a decision endorsed and spread to all the French colonies by the National Convention 6 months later. Toussaint Louverture, a black former slave who had been made Governor by France, re-established peace, fought off Spanish and British attempts to capture the island, and reestablished prosperity by daring measures. [2]... (following knowledge chunks omitted)
Prediction (with retrieved knowledge): Saint-Pierre et Miquelon (Saint-Pierre and Miquelon), a French overseas collectivity in the North American Atlantic Ocean.

Verbalized Knowledge: Toussaint L'Ouverture was a Haitian leader who led a successful revolt against French colonial rule in Haiti from 1791 to 1803. L'Ouverture was a former slave who became a prominent figure in the Haitian Revolution, which was a series of uprisings and battles fought between Haitian slaves and their French colonial oppressors. After years of fighting, L'Ouverture and his forces were able to secure Haiti's independence from France on January 1, 1804.
Prediction (with verbalized knowledge): Haiti (Saint-Domingue).

Neighbor Questions:
- The battle of Hastings in 1066 was fought in which country? (<Self> > <Wiki>)
- When Belgium declared its independence in 1830 it broke away from ___ control? (<Self> > <Wiki>)
- When did the French come to the new world? (<Self> > <Wiki>)
... (27 questions omitted)
$p_D(\texttt{<s>}|q)$: P(<Wiki>) = 24/30 = 0.8, P(<Self>) = 6/30 = 0.2.
$p_M(\texttt{<s>}|q)$: P(<Wiki>) = 0.576, P(<Self>) = 0.424.
Final Source Decision: <Self>

Figure 8: **Qualitative Example 2.** Example of SR-RAG inference where $p_D(\texttt{<s>}|q)$ corrects the source selection from $p_M(\texttt{<s>}|q)$. The model itself is selected as the knowledge source.

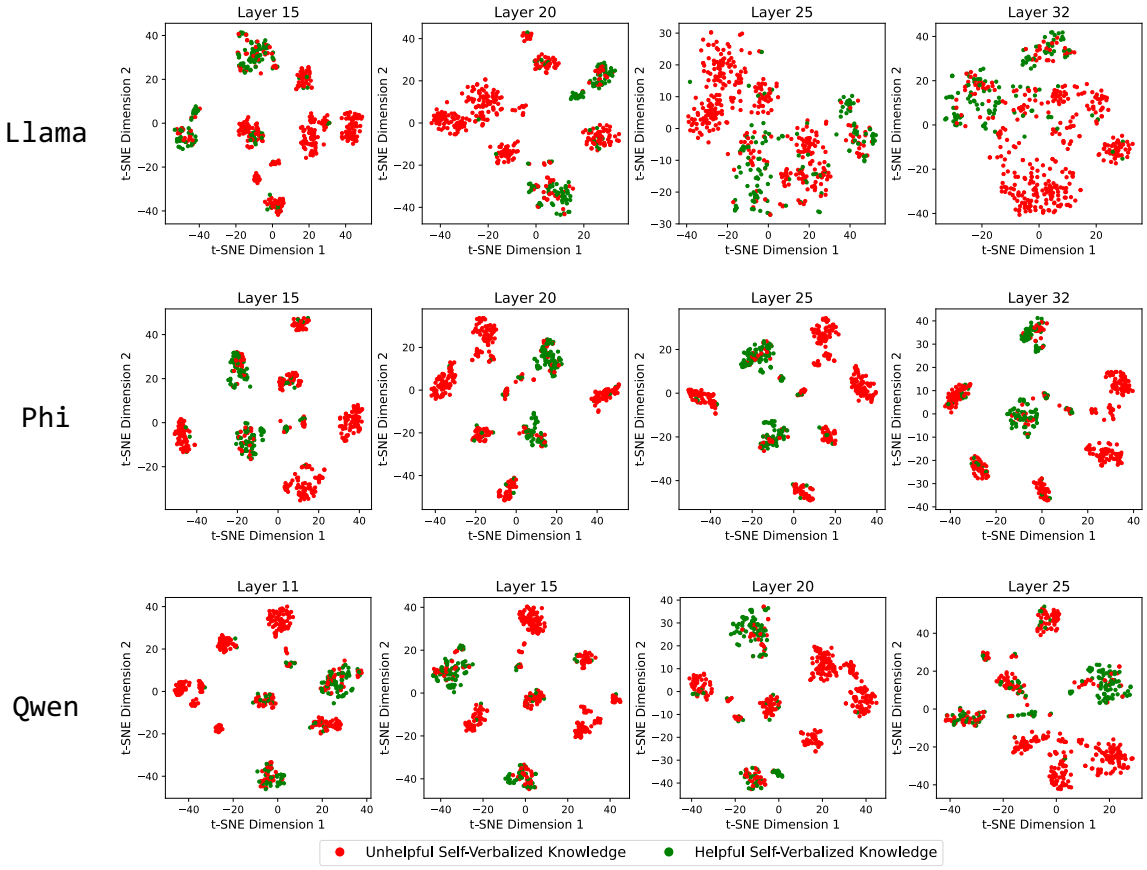


Figure 9: **Hidden State Visualization 1.** Hidden states of <EOQ> across three LLMs on PopQA, visualized using t-SNE. Llama = Llama-2-7B-Chat, Phi = Phi-3.5-Mini-Instruct, and Qwen = Qwen2.5-7B-Instruct.

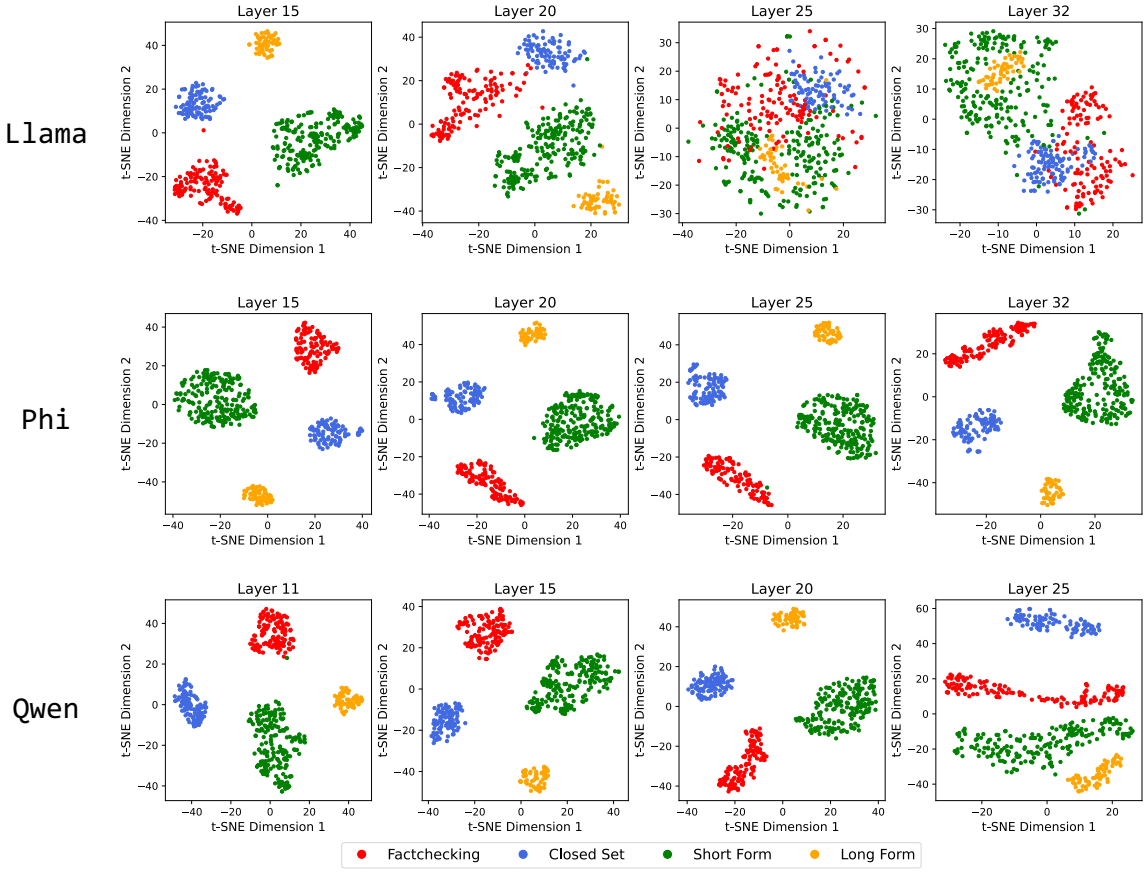


Figure 10: **Hidden State Visualization 2.** Hidden states of <EOQ> across three LLMs on a random sample of 500 data points in the SR-RAG training set, visualized using t-SNE. Llama = Llama-2-7B-Chat, Phi = Phi-3.5-Mini-Instruct, and Qwen = Qwen2.5-7B-Instruct.