

Probing-RAG: Self-Probing to Guide Language Models in Selective Document Retrieval

Ingeol Baek, Hwan Chang, Byeongjeong Kim, Jimin Lee, Hwanhee Lee[†]

Department of Artificial Intelligence, Chung-Ang University, Seoul, Korea
{ingeolbaek, hwanchang, michael97k, ljm1690, hwanheelee}@cau.ac.kr

Abstract

Retrieval-Augmented Generation (RAG) enhances language models by retrieving and incorporating relevant external knowledge. However, traditional retrieve-and-generate processes may not be optimized for real-world scenarios, where queries might require multiple retrieval steps or none at all. In this paper, we propose a Probing-RAG, which utilizes the hidden state representations from the intermediate layers of language models to adaptively determine the necessity of additional retrievals for a given query. By employing a pre-trained prober, Probing-RAG effectively captures the model's internal cognition, enabling reliable decision-making about retrieving external documents. Experimental results across five open-domain QA datasets demonstrate that Probing-RAG outperforms previous methods while reducing the number of redundant retrieval steps.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable performance across a variety of tasks (Brown, 2020; Ouyang et al., 2022; Touvron et al., 2023; Zhao et al., 2023). However, they still face challenges such as hallucinations (Ji et al., 2023; Zhang et al., 2023b; Huang et al., 2023) and factual errors (Min et al., 2023; Manakul et al., 2023; Wang et al., 2023a). Retrieval-Augmented Generation (RAG) leverages external knowledge related to the query through information retrieval steps to mitigate these issues.

Typically, the RAG pipeline follows a retrieve-and-generate process (Guu et al., 2020; Izacard and Grave, 2020; Singh et al., 2021; Izacard et al., 2023; Lazaridou et al., 2022; Shi et al., 2023), fetching relevant documents based on the user's input before generating a response. While this approach works well for simple single-hop QA (Ram et al., 2023), certain problems cannot be resolved in a single step

[†]Corresponding author.

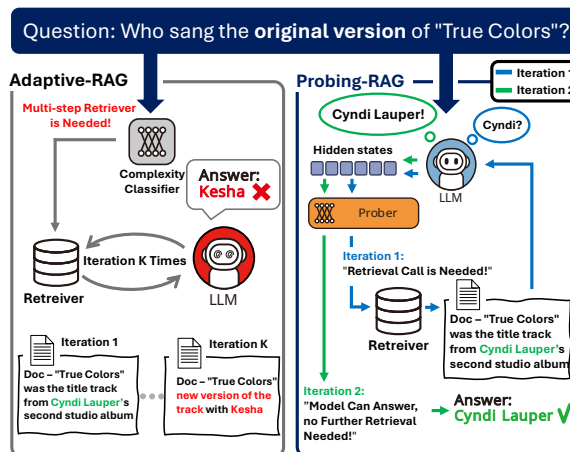


Figure 1: The left example illustrates how redundant retrieval steps, guided by an external query complexity classifier that does not reflect the LLM’s internal knowledge, can lead to wrong answers. In contrast, the right example shows that the model uses the prober to recognize that no further retrieval is needed, allowing it to generate the correct answer.

and require multiple retrieval steps (Zhang et al., 2023a; Shao et al., 2023; Cheng et al., 2024). On the other hand, some examples can be addressed using the LLM’s internal parametric knowledge without requiring any retrieval steps.

To optimize the RAG pipeline, recent research has introduced methods to adaptively adjust the number of retrievals. (Su et al., 2024; Mallen et al., 2022; Wang et al., 2023c). A notable example is Adaptive-RAG (Jeong et al., 2024), which classifies queries into three categories based on their complexity: no retrieval, single-step retrieval, or multi-step retrievals. However, as shown in Figure 1, external classifiers in Adaptive-RAG often fail to fully leverage the internal decision-making capabilities of the language model. This leads to unnecessary additional retrieval steps, resulting in knowledge conflicts between the model’s internal knowledge and externally retrieved information (Xie et al., 2024; Xu et al., 2024). Other

methods decide whether to retrieve based on token generation probabilities (Jiang et al., 2023b) and linguistic feedback (Zhang et al., 2024; Ding et al., 2024) from the LLM. However, these approaches still rely solely on the final output and cannot fully capture the model’s internal reasoning processes.

In this paper, we propose Probing-RAG, which determines whether the model needs to retrieve documents to answer a given question by examining the internal representations of the language model. As shown in the right example of Figure 1, Probing-RAG focuses on the hidden states of the LLM’s intermediate layers. These hidden states serve as inputs for the prober, which evaluates the need for further retrieval steps. In essence, the prober assesses whether the language model can generate an answer to a query using the available information. This approach ensures reliable decisions regarding the necessity of additional retrieval. We construct the prober by adding a fully connected layer to LLMs with a parameter size of just 5 MB, which is 2,000 times smaller than the external classification-based Adaptive-RAG (Jeong et al., 2024). To efficiently train the prober, we create a synthetic dataset derived from the widely used open-domain QA datasets.

We conduct experiments on five open-domain QA datasets to validate Probing-RAG. Experimental results demonstrate that Probing-RAG outperforms existing methods across both in-domain and out-of-domain datasets for the prober, reducing retrieval frequency by approximately 50% on average. Additionally, we provide a comprehensive analysis of prober training, including its position within the LLMs and the number of datasets required for training.

2 Related Work

Adaptive Retrieval Adaptive retrieval methods (Jiang et al., 2023b; Su et al., 2024; Asai et al., 2023) have emerged as an innovative approach that allows language models to actively decide when and what external information to retrieve based on the task type or the specific information contained within the query. These methods can be broadly categorized into three main approaches: external classifier based, LLM-based feedback, and confidence-based techniques.

External classifier based methods focus on training external models to optimize retrieval decisions based on query characteristics. For instance,

Adaptive-RAG (Jeong et al., 2024) incorporates an additional classifier that categorizes queries into three types: no retrieval, single-step retrieval, or multi-step retrieval. This classifier selects the optimal number of retrievals based on the complexity of each query. LLM-based feedback methods (Zhang et al., 2024; Ding et al., 2024) rely on the evaluation of response consistency to guide retrieval decisions. These methods use prompting or multilingual formulations to assess the quality of the model’s responses. If the LLM detects low consistency in its output, it triggers a retrieval to obtain relevant documents. Confidence-based approaches rely on the model’s token uncertainty to guide retrieval decisions. For example, FLARE (Jiang et al., 2023b) initiates retrieval when any token in a generated sentence has a probability below a certain threshold, while DRAGIN (Su et al., 2024) considers both token uncertainty and the attention weights between successive tokens to determine when to retrieve additional information.

Our work also aims to optimize RAG systems by dynamically adjusting the number of retrievals using a classifier similar to that of Adaptive-RAG. However, unlike Adaptive-RAG, which uses an external classifier, our study utilizes the internal representation of the model to effectively determine whether the model can solve the given query using the current context.

Knowledge Conflict In RAG systems, knowledge conflicts (Xu et al., 2024) can occur when external knowledge contradicts the parametric knowledge within LLMs, resulting in inconsistent outputs. Ideally, LLMs should identify these conflicts and provide distinct answers based on the conflicting contexts. However, LLMs often struggle to precisely detect and resolve such inconsistencies (Wang et al., 2023b). Instead, they often tend to overly rely on coherent and persuasive knowledge (Xie et al., 2024), which may not always reflect the most accurate or relevant information. This highlights the importance of using external knowledge only when the model is uncertain or lacks sufficient information.

Confidence Estimation via Internal State Estimating LLM’s factual confidence poses challenges when solely relying on outputs based on vocabulary distribution from the final layer (Mahaut et al., 2024). To address this, studies have focused on the outputs of the feed-forward neural networks and attention heads in the intermediate layers of

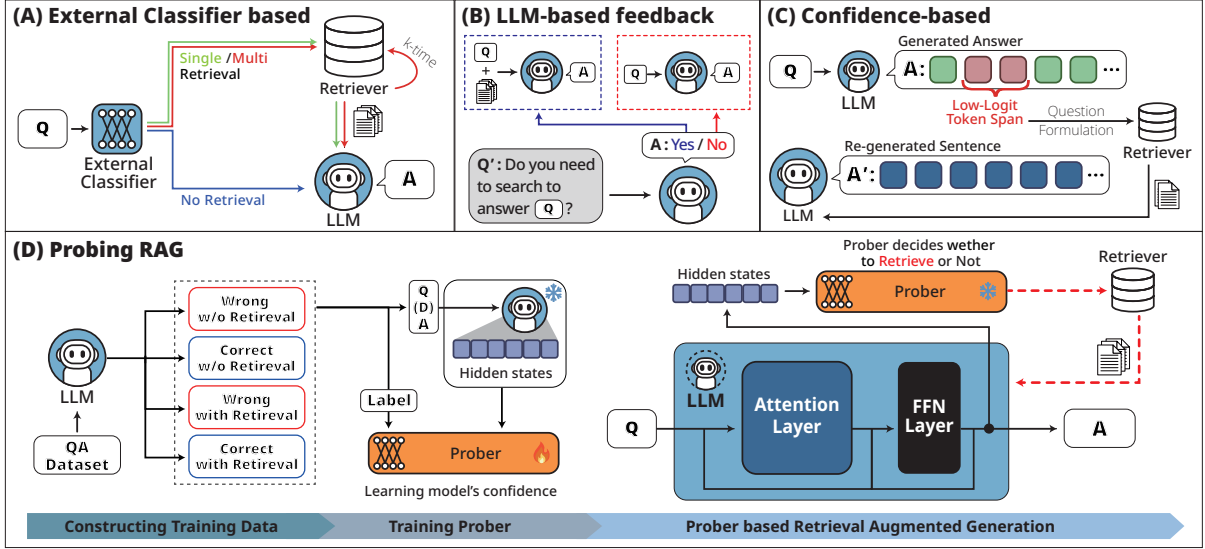


Figure 2: A conceptual comparison of various Adaptive-RAG approaches. (A) determines whether to perform retrieval based on query complexity measured by an external classifier. (B) decides retrieval based on the response from the LLM. (C) uses the confidence of the final token selection to determine retrieval. (D) Our proposed Probing-RAG decides retrieval using a prober model, which utilizes the internal hidden states of the LLM.

transformers. For summarization tasks, Lookback Lens (Chuang et al., 2024a) measures confidence by comparing attention weights on context versus newly generated tokens.

Recent work has expanded beyond confidence estimation to address hallucination mitigation using intermediate representations. Dola (Chuang et al., 2024b) contrasts logits from later and earlier layers to obtain a more reliable next-token distribution. Lookback Lens mitigates contextual hallucinations using a classifier-guided decoding strategy trained on attention-weight ratios. Building on this line of work, we propose a method to dynamically trigger retrieval by employing a prober trained on the LLM’s hidden states.

3 Method

We introduce Probing-RAG, an efficient RAG pipeline that incorporates a prober to determine whether the language model needs to retrieve additional documents. Similar to the conventional retrieval-augmented generation pipeline, our approach comprises a generating language model and a retriever. Different from the general pipeline, the generator of Probing-RAG leverages the output from the prober and adaptively calls the retriever based on the model’s internal hidden state.

3.1 Prober

Given the LLM’s hidden state during answer generation, the prober assesses whether an additional

retrieval step is necessary. We design the prober as a feed-forward network with a single hidden layer and an output layer for binary classification. Based on the findings of Chuang et al. (2024b), which indicate that lower layers in language models capture low-level information while higher layers capture more abstract, high-level information, we position the prober after the one-third point of the model to maximize the utility of these representations. In our experiments, we employ Gemma-2B (Team et al., 2024), an 18-layer model, as the generator in our RAG pipeline. Consequently, we position the prober on even-numbered layers starting from the 6th layer.

The prober utilizes the hidden states corresponding to the model-generated rationale (r) and answer (\hat{a}). Let H_{l_k} represent the hidden state at the k th layer, and let the combined length of the r and \hat{a} tokens be u . We define the hidden states corresponding to these tokens as:

$$T_{l_k} = H_{l_k}[-u :, d_{\text{model}}] \in \mathbb{R}^{u \times d_{\text{model}}} \quad (1)$$

To reduce the dimensionality of T'_{l_k} and obtain a summary representation, we first compute the mean of the hidden states across the token dimension, producing a single vector. Next, we normalize this vector to maintain numerical stability and consistency, resulting in the final input to the prober:

$$T'_{l_k} = \text{Norm}(\text{Mean}(T'_{l_k}, \text{dim} = 0)) \in \mathbb{R}^{d_{\text{model}}} \quad (2)$$

This normalized representation T'_{l_k} is then fed into the prober:

$$\text{logit}_{l_k} = \text{Prober}_{l_k}(T'_{l_k}) \quad (3)$$

Using the value of logit_{l_k} , prober decides whether or not to perform an additional retrieval.

	No Retrieval	With Retrieval
Correct	Query: What race track in the midwest hosts a 500 mile race every May? Rationale: The Indianapolis 500 is a 500-mile (800 km) automobile race... Pred: Indianapolis Motor Speedway Label: Indianapolis Motor Speedway ACC (Prober Training Label): 1	Query: Were Scott Derrickson and Ed Wood of the same nationality? Passages: {Retrieved passage 1, ..., Retrieved passage 5} Rationale: Scott Derrickson is an American film director, screenwriter... Pred: Yes Label: Yes ACC (Prober Training Label): 1
	Query: When was Poison's album "Shut Up, Make Love" released? Rationale: The album "Shut Up, Make Love" was released in ... Pred: 1990 Label: 2000 ACC (Prober Training Label): 0	Query: Are Random House Tower and 888 7th Avenue both used for real estate? Passages: {Retrieved passage 1, ..., Retrieved passage 5} Rationale: Random House Tower is a 40-story skyscraper in... Pred: Yes Label: No ACC (Prober Training Label): 0

Figure 3: Examples of prober training dataset.

3.2 Training Prober

To train the prober, we require pairs of $(T'_{l_k}, y)_1^N$, where T'_{l_k} is the input derived from the hidden states, and y is the output denoting whether additional retrieval is needed. To generate these pairs, we first use Chain-of-Thought (CoT) (Wei et al., 2022) prompting to let the LLM generate r and \hat{a} for the given question. The hidden state, T'_{l_k} , is recorded during this answer generation, and the label y is assigned based on the accuracy of \hat{a} compared to the ground truth. If the answer is correct, $y = 1$ (indicating no retrieval is needed); if the answer is incorrect, $y = 0$ (indicating retrieval is necessary).

For each question (q), two versions of r and \hat{a} are generated: one without retrieval and one with retrieval. This dual approach helps the prober learn to distinguish when retrieval improves performance and when it does not. By comparing hidden state representations across both retrieval and non-retrieval cases, the prober can better detect situations where retrieval is likely to improve accuracy. As shown in Figure 3, the dataset includes four types of examples: (with retrieval, $y = 0$),

Algorithm 1 Probing based RAG

Input: Q (question), D (corpus), d_j (retrieved j th document) \mathcal{M} (LLM), R (retriever)

Initialize: set count = 0, $\text{Logits} = \phi$

```

1: Define  $H_{l_k}$  = the hidden state at the pre-defined position
   of the  $k$ -th layer.
2: Define  $r, \hat{a}$  = the rationale and predicted answer tokens
   generated by  $\mathcal{M}$ .
3: Define  $L, l_k$  = the set of layers and each individual layer
   of  $\mathcal{M}$ .
4: Define  $\theta$  = the threshold.
5:  $(r, \hat{a}) \leftarrow \mathcal{M}.generate(q)$ 
6: while count < max iteration do
7:   if count is not 0 then
8:      $(r, \hat{a}) \leftarrow \mathcal{M}.generate(q, d_1, \dots, d_j)$ 
9:   end if
10:   $\text{Logits} \leftarrow \phi$ 
11:  for  $L \in \{l_6, l_8, \dots, l_k\}$  do
12:     $u = \text{len}(r) + \text{len}(\hat{a})$ 
13:     $T_{l_k} \leftarrow H_{l_k}([-u : d_{\text{model}}])$ 
14:     $T'_{l_k} \leftarrow \text{normalize}(\text{mean}(T_{l_k}, \text{dim} = 0))$ 
15:     $\text{logit}_{l_k} \leftarrow \text{Prober}_{l_k}(T'_{l_k})$ 
16:    Add  $\text{logit}_{l_k}$  to  $\text{Logits}$ 
17:  end for
18:  call retrieval, pass retrieval  $\leftarrow \text{Logits}[0], \text{Logits}[1]$ 
19:  if call retrieval +  $\theta >$  pass retrieval then
20:    if count is 0 then
21:       $d_1, \dots, d_j \leftarrow R(q, r, \hat{a})$ 
22:    else
23:       $d_1, \dots, d_j \leftarrow R(q, d_1, \dots, d_j, r, \hat{a})$ 
24:    end if
25:    count  $\leftarrow$  count + 1
26:  else
27:    break
28:  end if
29: end while

```

Output: \hat{a} .

(with retrieval, $y = 1$), (without retrieval, $y = 0$), and (without retrieval, $y = 1$).

We use three open-domain QA datasets to build the training dataset: HotpotQA (Yang et al., 2018), NaturalQA (Kwiatkowski et al., 2019), and TriviaQA (Joshi et al., 2017). To create a balanced dataset, we ensure an equal distribution of correctly and incorrectly answered questions. The final dataset consists of 26,060 training and 500 validation samples.

We train the probers using the generated dataset with a cross-entropy loss as follows:

$$L = -\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (4)$$

We present the hyperparameter details for training the prober in Appendix A.1.

3.3 Probing based Retrieval-Augmented Generation

In the first iteration, using CoT prompting, the LLM generates r and \hat{a} based solely on the input

question q :

$$(r, \hat{a}) = \begin{cases} LLM.generate(q) & \text{if iteration} = 0 \\ LLM.generate(q, \{d_k\}_1^j) & \text{otherwise} \end{cases} \quad (5)$$

After generating the initial r and \hat{a} , the prober assesses whether retrieval is necessary. To do this, we extract hidden state representations T'_{l_k} of the r and \hat{a} tokens from intermediate layers and feed them into the probers assigned to each layer to generate logit values. The decision to perform additional retrieval is based on the sum of the probers' logit values. If the difference between the logit for retrieval necessity and the logit indicating no need for retrieval is higher than the threshold θ , additional documents $d_{k_1}^j$ are retrieved based on the following conditions:

$$\{d_k\}_1^j = \begin{cases} R(q, r, \hat{a}) & \text{if iteration} = 0 \\ R(q, \{d_k\}_1^j, r, \hat{a}) & \text{otherwise} \end{cases} \quad (6)$$

In subsequent iterations, the LLM uses these retrieved documents along with q to generate updated versions of r and \hat{a} . This iterative process continues until either no further retrieval is needed or the maximum number of iterations is reached. Detailed information on procedure of the Probing-RAG is provided in Algorithm 1.

4 Experiment

4.1 Experimental Setup

Datasets We use five open-domain QA datasets for our experiments. We split the datasets used for training the prober as the in-domain dataset, while those not used for training are considered the out-of-domain dataset. The in-domain datasets include NaturalQA (NQ) (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and HotpotQA (Yang et al., 2018), and the out-of-domain datasets consist of MuSiQue (Trivedi et al., 2022) and 2WikimultihopQA (2Wiki) (Ho et al., 2020). To evaluate the performance of each Adaptive-RAG method, we sample 500 examples from the test set of each dataset.

Baselines We use the following baselines to evaluate the performance: **No Retrieval**, which proceeds directly based on the query without retrieval. **Single-step Approach**, which conducts a single retrieval call before QA. The **LLM-based** approach determines whether retrieval is necessary based on the language model's response to the question. **FLARE** (Jiang et al., 2023b) is a multi-round

retrieval-augmented method that triggers retrieval each time it encounters an uncertain token. When a sentence contains uncertain tokens, a query is generated to retrieve relevant passages. The sentence is then replaced with a newly generated one based on the query and the retrieved passage. **DRAGIN** (Su et al., 2024) determines when to retrieve based on token generation probabilities and performs query reformulation using attention weights. **Adaptive RAG** (Jeong et al., 2024) performs by fine-tuning a query complexity classifier. All methods use 4-shot examples, and we evaluate performance using Exact Match (EM) and Accuracy (ACC).

Implementation Details We use BM25 (Robertson and Jones, 1976; Robertson et al., 2009), a term-based sparse retrieval model, for all frameworks to ensure a fair comparison. We use the same document corpus and datasets as Jeong et al. (2024). For Adaptive-RAG, we train the classification model using the t5-large (Chung et al., 2024) model, following the approach of Jeong et al. (2024). We use the Gemma-2b (Team et al., 2024) and Mistral-7b (Jiang et al., 2023a) model as the QA model for all frameworks. We use the residual post (epoch=2) as the prober's hidden state representation position for Probing-RAG training and evaluation.

4.2 Main Results

We present the performance of each method on open-domain QA tasks in Table 1. We first observe that the single retrieval results outperform the no retrieval results in accuracy across all cases, except for TriviaQA and 2WikimultihopQA. We attribute the performance degradation when using the single-step approach on TriviaQA and 2WikimultihopQA to irrelevant external knowledge causing the decrease. We demonstrate that Probing-RAG demonstrates the best performance, with improvements of approximately 6.59% points and 8.35% points in ACC compared to the no-retrieval and single-step approaches, respectively. We also observe distinct characteristics in the results for each Adaptive-RAG method. We find that the FLARE method demonstrates lower performance compared to no retrieval across all datasets except for MuSiQue, and performs worse than the single-step approach across all datasets. For the FLARE, it continuously makes LLM calls to generate new queries and sentences that replace low-probability tokens, demonstrating its heavy reliance on model-generated text

Methods	HotpotQA		In-Domain NQ		TriviaQA		Out-of-Domain				Average	
							MuSiQue		2Wiki			
	EM	ACC	EM	ACC	EM	ACC	EM	ACC	EM	ACC	EM	ACC
Gemma-2b												
No Retrieval	16.8	28.0	15.0	24.6	37.4	45.4	3.2	4.8	22.6	<u>43.0</u>	19.0	<u>29.2</u>
Single-step Approach	14.6	<u>28.2</u>	11.4	26.0	19.6	38.8	1.8	<u>5.8</u>	22.8	38.4	14.0	27.4
Mistral-7b												
LLM-based	18.6	25.8	17.6	20.4	36.8	41.8	3.8	4.6	24.2	25.8	20.2	23.7
FLARE	13.2	21.0	9.0	21.8	13.8	31.0	1.2	5.0	21.6	27.8	11.8	21.3
DRAGIN	<u>19.8</u>	22.6	<u>18.8</u>	22.2	42.8	<u>47.0</u>	<u>4.2</u>	4.8	26.4	28.8	<u>22.4</u>	25.1
Adaptive-RAG	13.2	23.6	11.4	<u>26.2</u>	22.8	<u>40.8</u>	1.2	3.0	21.6	40.6	14.0	26.8
Probing-RAG(Ours)	21.8	39.4	21.6	35.0	<u>41.8</u>	52.2	4.8	8.8	<u>24.2</u>	43.6	22.8	35.8
Llama-3.1-8B												
No Retrieval	17.0	20.6	13.2	19.8	38.0	45.2	3.4	6.2	16.4	30.0	17.6	24.4
Single-step Approach	18.6	<u>34.2</u>	16.8	35.0	34.6	<u>51.0</u>	5.4	9.0	21.6	<u>32.6</u>	19.4	<u>32.4</u>
Llama-3.1-70B												
LLM-based	20.4	32.0	15.8	35.6	41.2	49.8	<u>5.6</u>	<u>9.4</u>	16.8	32.4	20.0	31.8
FLARE	20.4	32.0	15.4	34.4	35.0	45.6	4.4	6.6	18.6	31.0	18.8	29.9
DRAGIN	<u>21.2</u>	28.0	16.8	37.2	39.8	42.2	5.2	7.2	23.2	25.8	<u>21.2</u>	28.1
Adaptive-RAG	19.0	26.0	<u>17.2</u>	<u>37.4</u>	<u>40.8</u>	50.2	4.0	5.8	22.6	31.6	20.7	30.2
Probing-RAG(Ours)	22.4	38.6	20.8	39.4	43.2	52.2	5.8	9.8	<u>23.0</u>	33.4	23.0	34.7

Table 1: Experimental results on five different open-domain QA datasets. We indicate the highest performance in bold and underline the second highest.

	Total Retrieval Call	Retrieval Step Ratio		
		No	Single-step	Multi-step
LLM-based	2345	6.2%	93.8%	0.00%
FLARE	5317	12.41%	29.35%	58.24%
DRAGIN	13570	0.00%	1.20%	98.80%
Adaptive-RAG	3068	7.79%	61.96%	30.25%
Probing-RAG	1988	57.46%	20.19%	22.35%

Table 2: Distribution of retrieval steps using the Gemma-2b model, showing the proportion of instances with no, single-step, and multi-step retrieval.

and effectiveness primarily in large-parameter models. The DRAGIN shows lower accuracy compared to the single-step approach except for TriviaQA. However, as shown in Table 2, DRAGIN generally performs over five retrievals, which enhances exact match scores. We discuss DRAGIN’s performance in more detail in the case study in Table 5 and in Appendix C.3. Next, the Adaptive-RAG method outperforms the single-step approach on NQ, TriviaQA, and 2WikimultihopQA but shows lower results on the other datasets. Adaptive-RAG relies on an externally trained model to access query complexity, which fails to account for the internal knowledge of the QA model. We demonstrate that our proposed Probing-RAG outperforms all of these previous adaptive retrieval methods by avoiding redundant retrieval. By leveraging hidden state representations of LLMs, our approach leads to significant performance improvements.

4.3 Analysis

Number of Retrieval Steps In Table 2, we present the total number of retrievals for each

method along with the corresponding retrieval step ratio. We observe that LLM-based, FLARE, Adaptive-RAG, and DRAGIN perform 1.17, 2.67, 1.54, and 6.83 times more retrieval calls, respectively, compared to Probing-RAG. Probing-RAG performs retrieval calls when necessary, which is also reflected in the retrieval step ratio of Table 2. FLARE and Adaptive-RAG skip retrieval for 12.41% and 7.79% of cases and execute at least one retrieval for the rest. Moreover, DRAGIN performs multi-step retrieval in 98.8% of cases, resulting in computational overhead. However, as shown in Table 1, ACC performance is lower than Probing-RAG. Our method does not perform retrieval in 57.46% of cases. This demonstrates that the model successfully answers a higher proportion of queries without relying on retrieval, indicating that unnecessarily increasing the number of retrieval calls leads to computational overhead without effectively enhancing performance.

Size of Prober Training dataset To analyze the impact of training data size on performance in our framework, we conduct experiments by training the prober with varying amounts of data. We measure the performance on open-domain QA by training the prober on reduced datasets consisting of a sample of 1,000 data, as well as 1/4 and 1/2 proportions of the total 26,060 data points we present the result in Table 3. As expected, the overall results show that in most cases, performance improves as the amount of training data increases. How-

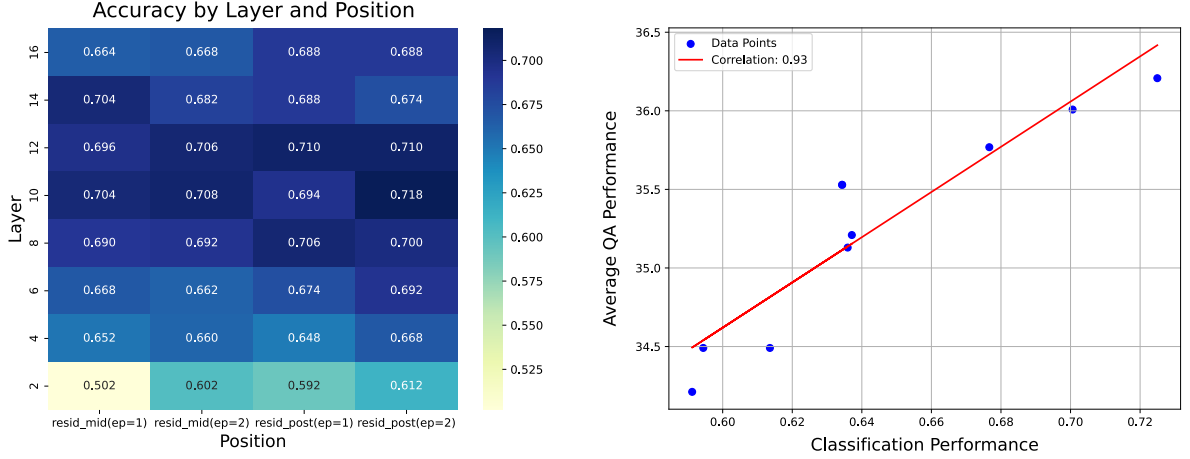


Figure 4: (Left) Probing accuracy measured for each model among the layer, (Right) Correlation between prober’s classification performance and QA performance, using the Gemma-2b model.

	In-domain	Out-of-domain	Total
FLARE	24.55	16.40	21.28
DRAGIN	30.61	16.77	25.07
Adaptive-RAG	30.15	21.76	26.79
Probing-RAG	42.18	26.15	35.77
1/2 of Train Set	41.32	25.35	34.93
1/4 of Train Set	40.65	26.35	34.93
1k Samples	40.32	25.75	34.49

Table 3: Performance comparison of variations in training data size for Probing-RAG and different retrieval methods using the Gemma-2b model. Values shown are accuracy (%) for in-domain, out-of-domain, and total averages.

ever, we also observe that the model trained with only 1k data points is comparable to the prober trained on one-third of the entire dataset in most of the datasets. Furthermore, we find that the prober trained on just 1k data points outperforms all of the previous methods, indicating that it is possible to effectively train the prober using a small dataset.

Answer Consistency among Retrieval Usage

We define model consistency as the ability of a model to correctly answer queries with retrieval methods that it previously answered correctly without retrieval. Specifically, low model consistency indicates situations where the model answers incorrectly due to the intervention of external knowledge, despite being capable of answering the query using only its internal knowledge. This phenomenon highlights the potential drawbacks of incorporating external information for queries that the model could originally handle independently. In Table 4, we evaluate how consistently single-step RAG, FLARE, DRAGIN, Adaptive-RAG, and

Consistency	HotpotQA	NQ	TriviaQA	MuSiQue	2Wiki
No Retrieval	100%	100%	100%	100%	100%
Single-step-RAG	73.8%	76.0%	82.0%	70.4%	85.0%
FLARE	74.4%	76.4%	83.8%	68.8%	72.8%
DRAGIN	75.2%	75.0%	81.8%	77.8%	69.8%
Adaptive-RAG	83.0%	76.6%	86.0%	74.6%	93.2%
Probing-RAG	90.6%	92.6%	91.0%	96.4%	96.6%

Table 4: Comparison of accuracy between no retrieval and adaptive retrieval augmentation to evaluate model consistency using the Gemma-2b model.

Probing-RAG answer queries solvable without retrieval. We observe that Probing-RAG achieves the highest consistency across all four datasets, each consisting of 500 examples. In contrast, as shown in Table 2, single-step RAG, FLARE, DRAGIN, and Adaptive-RAG perform retrieval for most queries. This leads to significant drops in consistency because incorrect retrieval timing and unnecessary external information cause the models to generate inconsistent answers.

4.4 Prober Accuracy

Accuracy among Layers We validate the performance of prober in classifying the necessity of retrieval calls using the test set. In Figure 4, the left side shows accuracy according to position and layer. Most probes achieve a performance near 0.7, with the highest performance observed at the residual post position. Therefore, we use the prober at this position for performance evaluation.

Correlation between Prober Accuracy and QA Performance We analyze the prober’s effectiveness by demonstrating the correlation between its classification performance and the average perfor-

Question & Label	Probing-RAG	DRAGIN
Question 1: Ezzard Charles was a world champion in which sport? Label: 'Fistfighting', 'Corner men', 'Corner persons', 'Prize-fighter', 'Boxing' ...	Rationale: Ezzard Charles was a world champion boxer , winning the world heavyweight title in 1949. He was born in ... Answer: Boxing ACC: 1, EM: 1 Total Number of Retrieval call: 0	Rationale: Ezzard Charles was a world champion in boxing . Answer: Boxing ACC: 1, EM: 1 Total Number of Retrieval call: 8
Question 2: Alfred Balk served as the secretary of the Committee on the Employment of Minority Groups in the News Media under which United States Vice Label: 'Nelson Rockefeller'	Rationale: The Committee on the Employment of Minority Groups in the News Media was established by the United States Vice President Nelson Rockefeller in 1972... Answer: Nelson Rockefeller ACC: 1, EM: 1 Total Number of Retrieval call: 3	Rationale: The National News Council was established by the Twentieth Century Fund in 1965. The committee was established to study... Answer: Lyndon B. Johnson ACC: 0, EM: 0 Total Number of Retrieval call: 8
Question 3: Leslie Nielsen trained in which of the armed services in WWII? Label: 'Royal Canadian'	Rationale: Leslie Nielsen was a Canadian actor, best known for his role as Dr. Strangelove in the 1964... Answer: Royal Canadian Air Force ACC: 1, EM: 0 Total Number of Retrieval call: 0	Rationale: Leslie Nielsen trained in the Royal Canadian . Answer: Royal Canadian ACC: 1, EM: 1 Total Number of Retrieval call: 6

Table 5: A case study using the Probing-RAG and DRAGIN methods. Accurate information referenced in the response is marked in **blue**, factual errors in **red**, and responses that are correct but not an exact match in **orange**.

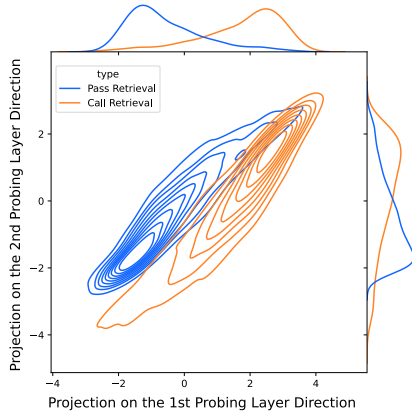


Figure 5: Kernel density estimate plot of logits using the Gemma-2b model, where orange indicates retrieval calls and blue indicates no retrieval needed. Marginal distributions are shown on the top and right. The results are projected onto the 10th and 12th residual post positions.

mance across five open-domain QA datasets, and present the results on the right of Figure 4. We utilize the prober classification and average performance obtained from varying thresholds in Table 8, along with those based on the number of probing layers in Table 9. As Figure 4 illustrates, improvements in the prober’s performance correspond to increases in the average, as indicated by a strong correlation of 0.93. Through this experiment, we demonstrate the effectiveness of the Probing-RAG approach, which utilizes trained probers that take the model’s internal hidden states as input.

Validity of Prober Training To demonstrate the validity of the prober trained on our constructed dataset and hidden states, we conduct kernel den-

sity estimation experiments. For the experiment, we extract the logit values of the prober based on the 500 test datasets created in Section 3.2. Additionally, we use the logits of the two best-performing probers at the residual post to create distributions for scenarios when retrieval is used and when it is not. In Figure 5, the overlap between the distributions of the generated logits is minimal; the logit values where retrieval is used are distributed toward the upper right, while those that do not require additional retrieval are distributed toward the lower left. This demonstrates that the prober, trained on the structure of our dataset as well as the hidden states of the rationale and answer tokens, effectively classifies the necessity of retrieval calls.

4.5 Case Study

We conduct a case study comparing the Probing-RAG and DRAGIN methods across three QA pairs from the TriviaQA and NQ datasets, with the results presented in Table 5. We measure the accuracy, exact match, total retrieval calls, and the rationale and answers generated by the model. In Question 1, Probing-RAG only uses the model’s internal knowledge to generate an answer without any retrieval. On the other hand, DRAGIN makes eight retrievals, even for questions that can be answered using the model’s internal knowledge, composes a single-sentence rationale, and then generates the final answer. For the Question 2, the prober determines that retrieval is necessary, conducts three retrieval steps, extracts the rationale, and provides an

answer. However, the DRAGIN method performs eight retrievals but generates a completely different answer. We demonstrate that repeated retrieval calls do not necessarily enhance performance and negatively affect it due to knowledge conflicts. In the final Question 3, our method provides the correct answer without retrieval but receives an exact match score of zero. In contrast, DRAGIN generates a concise single sentence and produces an accurate answer. Our analysis reveals that DRAGIN performs retrieval more than five times on average, generates a brief one-sentence rationale when forming answers, and achieves high exact match performance. This indicates that DRAGIN’s heavy reliance on external knowledge during answer generation contributes to improvements in exact match performance. In Appendix C.3, we provide a detailed comparison between ProbingRAG and DRAGIN on key factors such as the length of the generated answers, the number of retrieval calls, and the reasons for its high exact match scores.

5 Conclusion

In this study, we introduce Probing-RAG, an efficient adaptive retrieval pipeline that uses a pre-trained prober to determine whether additional retrieval is necessary by utilizing the language model’s hidden states. We introduce both the training dataset and the training method for the prober and show that Probing-RAG outperforms previous methods for various open-domain QA datasets. We also show that Probing-RAG reduces overhead through appropriate retrieval calls by effectively leveraging both external and internal knowledge via the prober.

Limitations

We propose a Probing-RAG method that reduces overhead by deciding whether to use retrieval based on the model’s hidden states and achieves high performance. However, the Probing-RAG method is limited to open-source LLMs, as it is incompatible with certain APIs where hidden state access is restricted. Additionally, our method involves elaborate data generation and some computational costs for training the prober. However, we also demonstrate that effective prober training is feasible with a small dataset. Furthermore, due to resource constraints, we were unable to test the prober on a broader range of models, including those with hyper-scale models such as 70B param-

eters, or validate its effectiveness on domain-specific datasets, which may limit its generalizability.

Ethics Statement

This study conducts QA tasks in the field of retrieval-augmented generation using language models. It also involves data generation and training. Therefore, it is important to recognize that language models may produce inappropriate responses. Additionally, as it may retrieve inappropriate content from the searched documents, developing management methods for this is essential. We believe this is a crucial area for future work.

Acknowledgement

This research was supported by Institute for Information & Communications Technology Planning & Evaluation (IITP) through the Korea government (MSIT) under Grant No. 2021-0-01341 (Artificial Intelligence Graduate School Program (Chung-Ang University)).

References

- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. [Self-RAG: Learning to retrieve, generate, and critique through self-reflection.](#) *arXiv preprint arXiv:2310.11511*.
- Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. 2024. Lift yourself up: Retrieval-augmented text generation with self-memory. *Advances in Neural Information Processing Systems*, 36.
- Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James Glass. 2024a. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. *arXiv preprint arXiv:2407.07071*.
- Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024b. [Dola: Decoding by contrasting layers improves factuality in large language models.](#) In *The Twelfth International Conference on Learning Representations*.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

- 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. *arXiv preprint arXiv:2402.10612*.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. [Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.
- Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.
- Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.
- Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. [Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7036–7050, Mexico City, Mexico. Association for Computational Linguistics.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. [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.
- 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.
- Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*.
- Matéo Mahaut, Laura Aina, Paula Czarowska, Momchil Hardalov, Thomas Müller, and Lluís Marquez. 2024. [Factual confidence of LLMs: on reliability and robustness of current estimators](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4554–4570, Bangkok, Thailand. Association for Computational Linguistics.
- Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*.
- Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.
- Sewon Min, Kalpesh Krishna, Xinxin Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. [FActScore: Fine-grained atomic evaluation of factual precision in long form text generation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- 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.

- Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.
- Stephen E Robertson and K Sparck Jones. 1976. Relevance weighting of search terms. *Journal of the American Society for Information science*, 27(3):129–146.
- 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. *arXiv preprint arXiv:2305.15294*.
- Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.
- Devendra Singh, Siva Reddy, Will Hamilton, Chris Dyer, and Dani Yogatama. 2021. End-to-end training of multi-document reader and retriever for open-domain question answering. *Advances in Neural Information Processing Systems*, 34:25968–25981.
- 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.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. ♪ **MuSiQue: Multi-hop questions via single-hop question composition**. *Transactions of the Association for Computational Linguistics*, 10:539–554.
- Cunxiang Wang, Xiaozhe Liu, Yuanhao Yue, Xian-gru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. 2023a. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*.
- Yike Wang, Shangbin Feng, Heng Wang, Weijia Shi, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. 2023b. Resolving knowledge conflicts in large language models. *arXiv preprint arXiv:2310.00935*.
- Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023c. Self-knowledge guided retrieval augmentation for large language models. *arXiv preprint arXiv:2310.05002*.
- 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*.
- Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. **Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts**. In *The Twelfth International Conference on Learning Representations*.
- Rongwu Xu, Zehan Qi, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. Knowledge conflicts for llms: A survey. *arXiv preprint arXiv:2403.08319*.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. **HotpotQA: A dataset for diverse, explainable multi-hop question answering**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Fengji Zhang, Bei Chen, Yue Zhang, Jacky Keung, Jin Liu, Daoguang Zan, Yi Mao, Jian-Guang Lou, and Weizhu Chen. 2023a. Repocoder: Repository-level code completion through iterative retrieval and generation. *arXiv preprint arXiv:2303.12570*.
- Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023b. Siren’s song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.
- Zihan Zhang, Meng Fang, and Ling Chen. 2024. **RetrievalQA: Assessing adaptive retrieval-augmented generation for short-form open-domain question answering**. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6963–6975, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

A Appendix

A.1 Hyperparameters

Name	Value
Learning rate	1e-3
Batch	12
Epoch	2
Dropout	0.1
Activation function	Silu
Noramlize	LayerNorm
Optimizer	AdamW
scheduler	ExponentialLR
gamma	0.995
GPUs	H100 \times 1

Table 6: Hyperparameters used for training the prober.

We present the detailed hyperparameters for training the prober model. We use the first epoch as a warm-up epoch, and in the second epoch, we create a checkpoint using early stopping based on prober’s classification accuracy.

B Prompt for Open-domain QA

For all methods, we use the same 4-shot prompt as in Table 7. When a retrieval call occurs, we include five retrieved passages in the Passages section. If directQA is required, we use the same prompt excluding the Passages section.

C Additional Analysis

C.1 Performance among Threshold of Prober

As mentioned in Section 3.3 and Algorithm 1, we add a threshold value to $Logit[0]$, which determines whether to perform a retrieval call. A lower threshold reduces the likelihood of retrieval, while a higher threshold increases the retrieval frequency. In Table 8, we measure performance by varying the threshold from -2 to 2 in increments of 1. For HotpotQA, NaturalQA, and IIRC, increasing the threshold leads to better performance, whereas for TriviaQA, MuSiQue, and 2Wikimul-tihop datasets, performance either decreases or remains similar. This demonstrates that simply increasing the amount of retrieval does not improve QA performance.

Query: Who was the first President of the United States?

Rationale: The United States was formed after gaining independence from Britain. The first President would have been elected soon after the formation of the country. George Washington is commonly known as the leader of the American Revolution and the first President.

Answer: George Washington

Query: Who wrote the play ‘Romeo and Juliet’?

Rationale: “Romeo and Juliet” is a famous play, a tragedy involving two young lovers. The play is widely associated with English literature from the Renaissance period. William Shakespeare is the most renowned playwright of the English Renaissance and is known for his tragedies.

Answer: William Shakespeare

Query: What is the main ingredient in traditional Italian pesto sauce?

Rationale: Pesto is a famous Italian sauce typically used in pasta dishes.

Traditional pesto originates from the Liguria region, specifically Genoa. The primary ingredients include basil, olive oil, garlic, pine nuts, and Parmesan cheese. The main flavoring herb that distinguishes pesto is basil.

Answer: Basil

Question: Which writer was from England, Henry Roth or Robert Erskine Childers?

Rationale: Henry Roth was an American novelist, best known for his novel *Call It Sleep* (1934). He was born in Austria-Hungary (now Ukraine) and emigrated to the United States as a child. Robert Erskine Childers was an English-born writer and Irish nationalist. He was born in London, England, in 1870 and is best known for his novel *The Riddle of the Sands* (1903). Childers later became involved in Irish politics and was a prominent figure in the Irish independence movement.

Answer: Robert Erskine

Passages:

{Retrieved Passages}

Question:

{New Query}

Rationale:

Table 7: The 4-shot prompt used for Adaptive RAG methods.

Threshold (θ)	HotpotQA	NQ	TriviaQA	MuSiQue	2Wiki
-2	37.72	32.33	47.90	8.78	44.31
-1	39.12	34.53	49.70	9.98	44.31
0	39.12	35.53	50.50	9.98	43.71
1	40.12	36.53	51.50	9.58	42.32
2	40.72	38.92	50.70	9.38	41.31

Table 8: Results among varying the classifying threshold for prober.

Probing Layer	HotpotQA	NQ	TriviaQA	MuSiQue	2Wiki
6	38.12	36.72	50.70	8.18	42.32
6, 8	38.12	33.13	48.90	8.78	43.51
6, 8, 10	38.12	32.34	48.70	8.98	44.31
6, 8, 10, 12	38.92	33.73	49.30	9.38	44.31
6, 8, 10, 12, 14	39.12	34.53	49.70	9.98	44.31

Table 9: Accuracy among the prober location within the layers.

C.2 Impact of Varying the Number of Prober Layers on QA Performance

We use the prober’s logits to decide whether to perform retrieval. Summing the logits from each layer’s prober provides a soft voting effect and enhances model performance, as empirically shown in Table 9. We measure performance while removing one probing layer at a time. We observe that having more probers increases performance across most datasets except for NaturalQA and TriviaQA. Using probers up to layer 14 improves the average performance across most datasets by 2.0 percentage points compared to using a single prober at layer 6.

C.3 Depth-wise Comparison between Probing-RAG and DRAGIN

In Figure 6, the numbers above the bar plots represent the average number of retrieval calls. The upper left shows the number of rationale sentences when Probing-RAG answers correctly, and the lower left shows the number when Probing-RAG answers incorrectly. The upper-right displays the number of rationale sentences when DRAGIN answers correctly, and the lower-right shows the number when DRAGIN answers incorrectly. As shown in Table 1, DRAGIN has lower accuracy but higher exact match performance. To explain this, we compare the outputs of DRAGIN and Probing-RAG in the case study presented in Table 5. For a more substantial explanation, we analyze statistically the answers generated by DRAGIN and Probing-RAG to understand the reason behind the high exact match performance. The top-left and top-right of Figure 6

shows the number of sentences in the rationale sections for Probing-RAG and DRAGIN, respectively. Probing-RAG makes 0.6 and 0.26 retrieval calls when generating one and two sentences, respectively. In contrast, DRAGIN makes 5.66 and 5.63 retrieval calls. Making more than five retrieval calls to generate a single sentence enables accurate answers based on external knowledge. As shown in Table 5, DRAGIN directly references external knowledge to produce its responses. DRAGIN ignores questions that the language model could answer using internal knowledge and heavily relies on external information. This reliance leads to decreased accuracy when external knowledge is inaccessible. In contrast, Probing-RAG makes fewer than one retrieval call, generating correct answers without depending on external knowledge for questions answerable through internal knowledge. This approach decreases exact match performance but increases accuracy when the model uses its internal knowledge to generate answers. We also observe similar sentence lengths and numbers of retrieval calls even when incorrect answers are generated. Additionally, Adaptive-RAG methods typically perform more retrievals for complex questions. However, DRAGIN makes fewer retrieval calls when it produces incorrect answers than when it answers correctly. In contrast, Probing-RAG makes more retrieval calls in situations where it generates incorrect answers. Through the statistical analysis in Figure 6, we explain why DRAGIN exhibits high Exact Match performance but low accuracy. DRAGIN relies heavily on external knowledge by making multiple inefficient retrieval calls. In contrast, Probing-RAG significantly improves accuracy by efficiently utilizing retrieval and leveraging internal knowledge.

C.4 Additional Case Study

To enhance the case study section, we add two examples from each of the four datasets used in our evaluation: NaturalQA, HotpotQA, TriviaQA, and MuSiQue. We provide the rationale and answers generated by the Probing RAG, Adaptive RAG, DRAGIN, and FLARE methods, along with their accuracy and number of retrievals. Here, an accuracy of 1 indicates a correct answer, and 0 indicates an incorrect one.

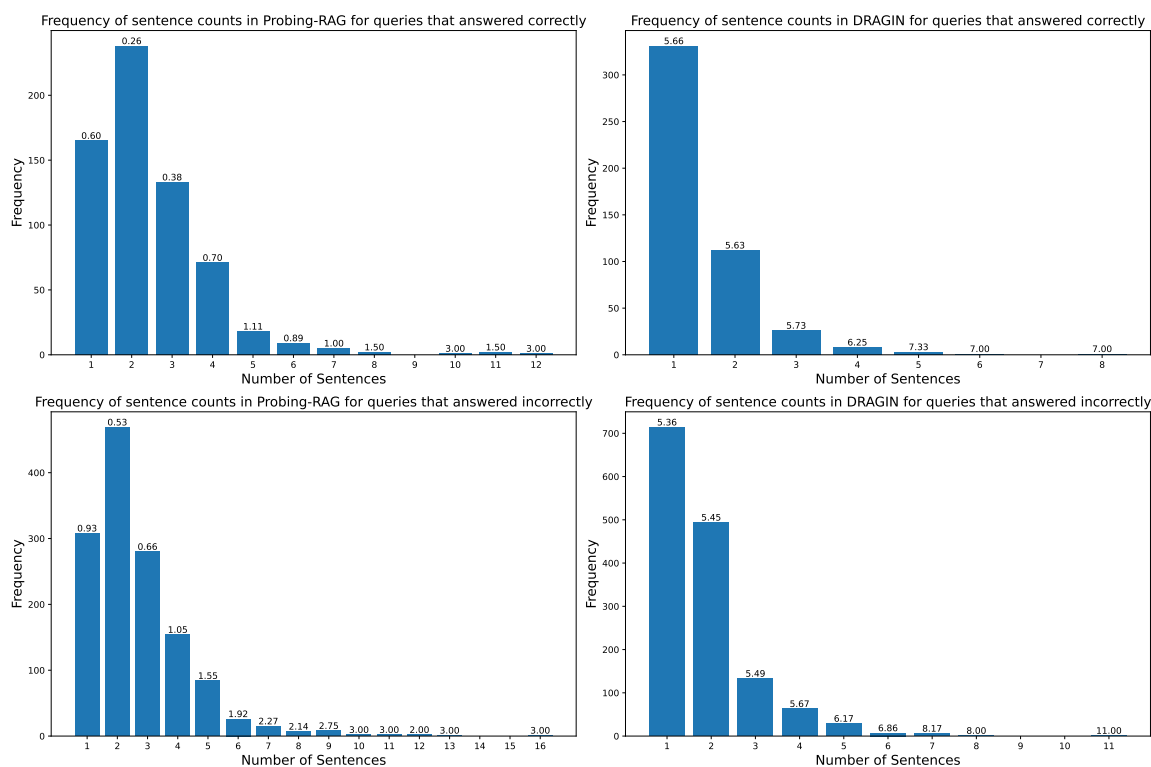


Figure 6: The numbers above the bar plots represent the average number of retrieval calls. The top-left section shows the number of rationale sentences when Probing-RAG answered correctly, and the bottom-left shows when it answered incorrectly. The top-right indicates the number of rationale sentences when DRAGIN answered correctly, while the bottom-right indicates when it answered incorrectly.

NaturalQA Example 1	
Question: which apostle had a thorn in his side	
Answer: Paul	
Probing-RAG	Rationale: The apostle Paul had a thorn in his side. This is a reference to 2 Corinthians 12:7-10. Answer: Paul Accuracy: 1 Number of Total Retrieval Call: 0
Adaptive-RAG	Rationale: Paul the Apostle was a prominent figure in the early Christian church. He was a follower of Jesus Christ and was known for his missionary work. He was also known for his writings in the New Testament, particularly his letters to the early Christian communities. Paul was a prolific writer and is considered one of the most influential figures in the history of Christianity. He is known for his teachings on topics such as salvation, grace, and the Christian life. Answer: Paul the Apostle Accuracy: 1 Number of Total Retrieval Call: 1
DRAGIN	Rationale: Paul the Apostle had a thorn in his side. This is stated in 2 Corinthians 12:7-9. Answer: Paul the Apostle Accuracy: 1 Number of Total Retrieval Call: 6
FLARE	Rationale: Paul the Apostle was a prominent figure in the early Christian church. He was a follower of Jesus Christ and was known for his missionary work. He was also known for his writings in the New Testament, particularly his letters to the early Christian communities. Paul was a prolific writer and is considered one of the most influential figures in the history of Christianity. He is known for his teachings on topics such as salvation, grace, and the Christian life. Answer: Paul Accuracy: 1 Number of Total Retrieval Call: 5
NaturalQA Example 2	
Question: who sang the original version of true colors	
Answer: Cyndi Lauper	
Probing-RAG	Rationale: The original version of "True Colors" was sung by Cyndi Lauper. The song was released in 1986 and became a hit, reaching number one on the Billboard Hot 100 chart. Answer: Cyndi Lauper Accuracy: 1 Number of Total Retrieval Call: 0
Adaptive-RAG	Rationale: True Colors is a song by Russian-German electronic music producer Zedd. The original version of the song featured uncredited vocals by Tim James and was included on Zedd's second studio album, "True Colors" (2015). The official single is a new version of the track with vocals provided by the American singer Kesha and was released as the album's fourth single on April 29, 2016. They performed the song live at Coachella 2016. The song appeared in "FIFA 17". "True Colors" was Kesha's first music release in over three years, as she has been locked. Answer: Kesha Accuracy: 0 Number of Total Retrieval Call: 1
DRAGIN	Rationale: The original version of "True Colors" was sung by Cyndi Lauper. Answer: Cyndi Lauper Accuracy: 1 Number of Total Retrieval Call: 7
FLARE	Rationale: The song was released on 29 October 2010. Answer: Kesha Accuracy: 0 Number of Total Retrieval Call: 1

Table 10: Output examples for the NaturalQA dataset.

HotpotQA Example 1	
Question: Which dog’s ancestors include Gordon and Irish Setters: the Manchester Terrier or the Scotch Collie?	
Answer: Scotch Collie	
Probing-RAG	Rationale: The Manchester Terrier is a breed of dog that originated in England. It is a small, sturdy dog with a short, smooth coat. The Scotch Collie is a breed of dog that originated in Scotland. It is a medium-sized dog with a long, straight coat. Answer: Scotch Collie Accuracy: 1 Number of Total Retrieval Call: 0
Adaptive-RAG	Rationale: The Manchester Terrier is a breed of dog that originated in England. It is a small, sturdy dog with a short, smooth coat. The Manchester Terrier is known for its intelligence, loyalty, and playful nature. The Scotch Collie is a breed of dog that originated in Scotland. It is a medium-sized dog with a long, smooth coat. The Scotch Collie is known for its intelligence, loyalty, and playful nature. Answer: Manchester Terrier Accuracy: 0 Number of Total Retrieval Call: 2
DRAGIN	Rationale: The Manchester Terrier is a breed of dog of the smooth-haired terrier type. The Scotch Collie is a breed of dog, categorized as a terrier. Answer: Scotch Collie Accuracy: 1 Number of Total Retrieval Call: 5
FLARE	Rationale: The Manchester Terrier is a breed of dog of the smooth-haired terrier type. The Irish Red and White Setter is a medium-sized dog with a long, silky coat. The Irish Setter is a medium-sized dog with a long, silky coat. The Scotch Collie is a landrace breed of dog which originated from the highland regions of Scotland. The Manchester Terrier is a breed of dog of the smooth-haired terrier type. Answer: Scotch Collie Accuracy: 1 Number of Total Retrieval Call: 5
HotpotQA Example 2	
Question: Alfred Balk served as the secretary of the Committee on the Employment of Minority Groups in the News Media under which United States Vice	
Answer: Nelson Rockefeller	
Probing-RAG	Rationale: The Committee on the Employment of Minority Groups in the News Media was established by the United States Vice President Nelson Rockefeller in 1972. The committee was established to study the employment of minorities in the news media. The committee was chaired by Alfred Balk. Answer: Nelson Rockefeller Accuracy: 1 Number of Total Retrieval Call: 3
Adaptive-RAG	Rationale: Alfred Balk was a United States Vice President. He served as the Secretary of the Committee on the Employment of Minority Groups in the News Media under President Lyndon B. Johnson. Answer: Lyndon B. Johnson. Answer: Lyndon B. Johnson Accuracy: 0 Number of Total Retrieval Call: 2
DRAGIN	Rationale: The National News Council was established by the Twentieth Century Fund in 1965. The committee was established to study the employment of minorities in the news media. Alfred Balk was the secretary of the committee. Answer: Lyndon B. Johnson. Answer: Lyndon B. Johnson Accuracy: 0 Number of Total Retrieval Call: 8
FLARE	Rationale: The Vice President of the United States is the second-highest-ranking official in the United States federal government, after the President of the United States. The Vice President is the President of the Senate and is the presiding officer of the Senate. The Vice President is the President of the United States in the event of the death, resignation, or removal in office of the President. Answer: Lyndon B. Johnson Accuracy: 0 Number of Total Retrieval Call: 3

Table 11: Output examples for the HotpotQA dataset.

TriviaQA Example 1	
Question: Which ship was sunk by a German submarine in 1915 with a loss of 1400 lives including 128 Americans?	
Answer: [Roman Lusitania, Luso-, Lusitânia, Lusitania, Hispania Lusitania, Lusitania (ancient region), Roman province Lusitania, Lusitania (Roman province)]	
Probing-RAG	<p>Rationale: The Lusitania was a British passenger liner that was sunk by a German submarine in 1915. The Lusitania was carrying 128 Americans, including 128 Americans, and 128 British passengers. The Lusitania was sunk by a German submarine, and the loss of life was a major factor in the United States' decision to enter World War I.</p> <p>Answer: Lusitania</p> <p>Accuracy: 1</p> <p>Number of Total Retrieval Call: 0</p>
Adaptive-RAG	<p>Rationale: The Lusitania was a British passenger liner that was sunk by a German submarine in 1915. The Lusitania was carrying 1,959 passengers and crew, including 128 Americans. The Lusitania was torpedoed by the German submarine U-20, killing 1,198 people, including 128 Americans. The Lusitania was the first major loss of life in the First World War.</p> <p>Answer: Lusitania</p> <p>Accuracy: 1</p> <p>Number of Total Retrieval Call: 1</p>
DRAGIN	<p>Rationale: The RMS Lusitania was a British passenger liner that was sunk by a German submarine in 1915. The Lusitania was the largest passenger ship in the world at the time of its sinking. The Lusitania was carrying 1,959 passengers and crew when it was sunk. The Lusitania was the largest passenger ship in the world at the time of its sinking.</p> <p>Answer: Lusitania</p> <p>Accuracy: 1</p> <p>Number of Total Retrieval Call: 5</p>
FLARE	<p>Rationale: The Lusitania was a British passenger liner that was sunk by a German submarine in 1915. The sinking of the "Lusitania" was the event that decisively swung American opinion against Germany. The sinking of the "Lusitania" was the event that decisively swung American opinion against Germany.</p> <p>Answer: Lusitania</p> <p>Accuracy: 1</p> <p>Number of Total Retrieval Call: 3</p>
TriviaQA Example 2	
Question: Balice international airport is in which country?	
Answer: [Etymology of Poland, Polish state, Poleand, Rzeczpospolita Polska, Polland, ISO 3166-1:PL, Polskor, Republic of Poland, POLAND, Fourth Poland, Country- Poland, Poleland, Polska, Poland, Poland., Pols Land, Polnd, Po land, Lenkija, Bastarnia, Pologne, PolanD, Polija, Pole land, Architecture of Poland, Ploand]	
Probing-RAG	<p>Rationale: Balice international airport is in the country of Croatia.</p> <p>Answer: Croatia</p> <p>Accuracy: 0</p> <p>Number of Total Retrieval Call: 0</p>
Adaptive-RAG	<p>Rationale: The airport is located in the village of Balice, west of the city centre, in southern Poland.</p> <p>Answer: Poland</p> <p>Accuracy: 1</p> <p>Number of Retrieval Call: 1</p>
DRAGIN	<p>Rationale: Balice international airport is in Poland.</p> <p>Answer: Poland</p> <p>Accuracy: 1</p> <p>Number of Total Retrieval Call: 7</p>
FLARE	<p>Rationale: The airport is located in the village of Balice, west of the city centre, in southern Poland.</p> <p>Answer: Poland</p> <p>Accuracy: 1</p> <p>Number of Total Retrieval Call: 1</p>

Table 12: Output examples for the TriviaQA dataset.

MuSiQue Example 1	
Question: Who wrote "Turn Me On" by performer of "Happy Pills"?	
Answer: John D. Loudermilk	
Probing-RAG	Rationale: "Turn Me On" is a song by Norah Jones from her album First Sessions / Come Away with Me. The song was written by John D. Loudermilk. Answer: John D. Loudermilk Accuracy: 1 Number of Total Retrieval Call: 1
Adaptive-RAG	Rationale: "Turn Me On" is a song by Norah Jones from her album First Sessions / Come Away with Me. The song was written by John D. Loudermilk. Answer: John D. Loudermilk Accuracy: 1 Number of Total Retrieval Call: 1
DRAGIN	Rationale: The performer of "Happy Pills" is the American singer-songwriter and actress, Cher. She is best known for her work in the 1960s and 1970s, and for her work in the 1980s and 1990s. She has sold over 100 million records worldwide, making her one of the best-selling music artists of all time. Answer: Cher Accuracy: 0 Number of Total Retrieval Call: 4
FLARE	Rationale: The Beatles' first album, Please Please Me, was released in the United Kingdom on 22 March 1963. The Beatles' White Album was released in the United States on 15 April 1963. The first episode of the series aired on 17 December 1999. Answer: The Beatles Accuracy: 0 Number of Total Retrieval Call: 3
MuSiQue Example 2	
Question: What is the date of birth of the person who was part of Ratata?	
Answer: 11 September 1962	
Probing-RAG	Rationale: Ratata was a pop group formed in 1980. The group was composed of Mauro Scocco and Johan Ekelund. The group was active from 1980 to 1989. Answer: 11 September 1962 Accuracy: 1 Number of Total Retrieval Call: 1
Adaptive-RAG	Rationale: Ratata was a Polish resistance organization that was active during World War II. The organization was founded in 1942 by Augustyn Träger (25 August 1896 – 22 April 1957), codenames "Sek" (Knot) and "Tragarz" (Porter), and Roman Träger (1922–1944). The organization was active in the Polish resistance movement in the occupied territories of Poland. Answer: 25 August 1896 Accuracy: 0 Number of Total Retrieval Call: 2
DRAGIN	Rationale: The answer is Peter Fliesteden. Answer: Peter Fliesteden Accuracy: 0 Number of Total Retrieval Call: 6
FLARE	Rationale: Ratata was a pop group formed in 1980. The group was composed of Mauro Scocco and Johan Ekelund. The group was active from 1980 to 1989. Answer: 11 September 1962 Accuracy: 1 Number of Total Retrieval Call: 3

Table 13: Output examples for the MuSiQue dataset.