Dynamic Uncertainty Ranking: Enhancing Retrieval-Augmented In-Context Learning for Long-Tail Knowledge in LLMs

Shuyang Yu[1]*, **Runxue Bao**[2]†, **Parminder Bhatia**[2],
Taha Kass-Hout[2], **Jiayu Zhou**[3], **Cao Xiao**[2]†
[1]Department of Computer Science and Engineering, Michigan State University
[2]GE Healthcare
[3]School of Information, University of Michigan

Abstract

Large language models (LLMs) can learn vast amounts of knowledge from diverse domains during pre-training. However, long-tail knowledge from specialized domains is often scarce and underrepresented, rarely appearing in the models' memorization. Prior work has shown that in-context learning (ICL) with retriever augmentation can help LLMs better capture long-tail knowledge, reducing their reliance on pre-trained data. Despite these advances, we observe that LLM predictions for long-tail questions remain uncertain to variations in retrieved samples. To take advantage of the uncertainty in ICL for guiding LLM predictions toward correct answers on long-tail samples, we propose a reinforcement learning-based dynamic uncertainty ranking method for retrieval-augmented ICL that accounts for the varying impact of each retrieved sample on LLM predictions. Our approach prioritizes more informative and stable samples while demoting misleading ones, updating rankings based on the feedback from the LLM w.r.t. each retrieved sample. To enhance training efficiency and reduce query costs, we introduce a learnable dynamic ranking threshold, adjusted when the model encounters negative prediction shifts. Experimental results on various question-answering datasets from different domains show that our method outperforms the best baseline by 2.76%, with a notable 5.96% boost in accuracy on long-tail questions that elude zero-shot inference. Our code is available at https://github.com/Yu-shuyan/uncertian_ranker.

1 Introduction

Pretrained large language models (Brown, 2020; Touvron et al., 2023; Almazrouei et al., 2023) have achieved remarkable success across various natural language processing (NLP) tasks, such as summarization (Zhang et al., 2019; Van Veen et al., 2024), question answering (Jiang et al., 2021; Wang et al., 2024a), and code generation (Li et al., 2023b; Wang et al., 2024b). These impressive results are largely due to their pre-training on vast, web-sourced datasets spanning multiple domains. However, these real-world datasets often follow a long-tail distribution (Liu et al., 2019; Mallen et al., 2022; Dai et al., 2023; Sun et al., 2023a), where knowledge from less frequent domains is underrepresented. Consequently, certain domain-specific information may be rarely or even never included in the LLMs' memorization (Kandpal et al., 2023). As a result, LLMs struggle to provide accurate responses to queries drawn from these long-tail distributions, since the pre-training process fails to capture this sparse information.

In-context learning (ICL) (Brown, 2020) is a few-shot learning method that queries LLMs by concatenating relevant samples with the test query, without updating the model's parameters. Kandpal et al. (2023) found that ICL, when combined with retriever augmentation, can reduce LLMs' reliance on pre-training knowledge by retrieving relevant examples related to long-tail queries during inference. Common retrieval methods used to select augmentation examples for ICL include random selection (Wei et al., 2022; Wang et al., 2022), off-the-shelf retrievers (e.g., BM25 (Robertson et al., 2009)), and fine-tuned retrievers (e.g., PromptPG (Lu et al., 2022)). However, prior works (Zhao et al., 2021; Liu et al., 2021; Lu et al., 2021; Chen et al., 2023) have shown that ICL with different selection and ordering of the retrieved samples could lead to unstable predictions of LLMs. In our experiments, we observed a similar pattern: when utilizing existing methods to retrieve relevant samples for ICL, the model's predictions for long-tail questions—those not captured by zero-shot inference—exhibited particularly high uncertainty. In some cases, a subset of the retrieved samples led

*Work was done during the internship at GE Healthcare.
†Correspondence to: Runxue Bao, Cao Xiao <{runxue.bao, cao.xiao}@gehealthcare.com>

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 8985–8997

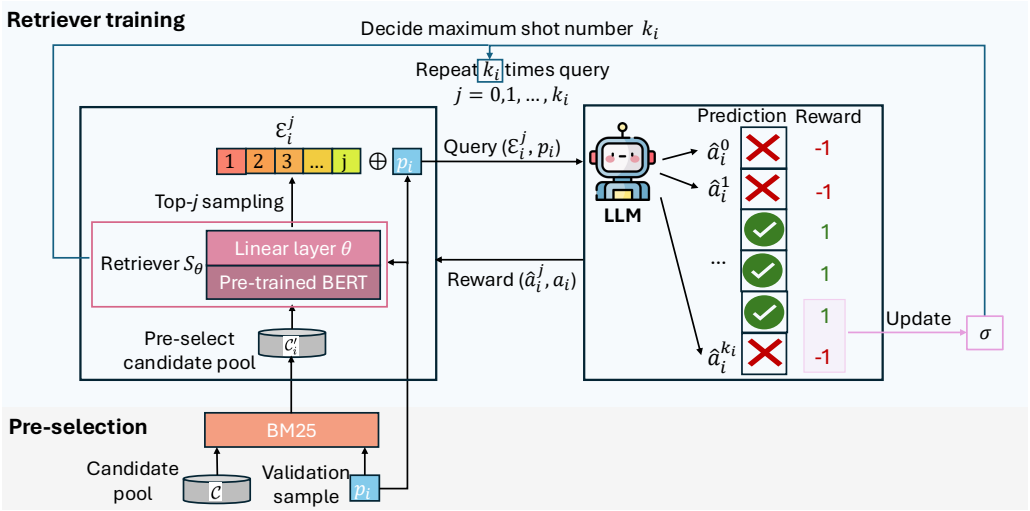


Figure 1: Training framework of the proposed method. After pre-selection using BM25 for each validation sample p_i, we conduct from 0-shot to k_i-shot inference and update retriever S_θ according to the dynamic impacts of each sample on LLMs based on the reward from LLM. To reduce the query cost, we update the threshold σ when the LLM experiences a negative prediction change. The query time k_i is decided by retriever score S_θ and threshold σ.

to correct predictions, while the full set misled the model, even with the same retrieval method.

In this paper, to enhance the retrieval augmentation for long-tail samples regarding LLM's uncertainty, we propose a reinforcement learning-based dynamic uncertainty ranking method motivated by reinforcement learning's capacity to search for optimal retrieved samples based on the LLM's feedback (Lu et al., 2022). Specifically, our approach trains a retriever to prioritize informative and stable samples while down-ranking misleading ones, enhancing performance on both head and tail distributions. We build on the BERT-based retriever architecture (Devlin, 2018) with an appended linear layer. During the training of the retriever, only the linear layer is fine-tuned. Initially, BM25 (Robertson et al., 2009) is used for pre-selection, and the retriever is trained using policy gradients (Sutton et al., 1999), guided by feedback from the LLM for each retrieved sample. To improve efficiency, we introduce a learnable dynamic threshold as a budget controller for retrieval, selecting only samples with high-ranking scores above this threshold, which adjusts whenever the LLM experiences a negative prediction change, i.e., the prediction changes from true to false. To evaluate the proposed approach, we compared our method with the state-of-the-art methods across both multi-choice and open-ended question-answering (QA) datasets from different domains. The experimental results show that our method outperforms the best baseline by 2.76%. Long-tail questions failed to be captured by a zero-shot inference benefit particularly from our proposed method. The accuracy of long-tail questions of our method surpasses previous methods with a large margin of up to 5.96%.

We summarize our key contributions as follows:

- We investigate the limitations of existing retrieval-augmented ICL approaches for handling long-tail questions, highlighting how variations in retrieved samples contribute to prediction uncertainty.
- We propose a reinforcement learning-based dynamic uncertainty ranking method with a budget controller that considers the dynamic impact of each retrieved sample on the LLM's prediction, which selectively elevates informative retrieved samples and suppresses misleading ones with minimal query costs.
- Extensive experiments demonstrate that our method consistently outperforms the state-of-art method on multiple QA datasets from different domains, achieving nearly a 6% improvement in accuracy for long-tail questions.

2 Related Work

In-context learning (ICL). ICL (Brown, 2020) queries the LLMs with a concatenation of related samples and the test query without parameter updating. To improve the quality of ICL, retrievers have been proposed to select related samples, which can

be categorized into sparse retrievers (e.g. (Robertson et al., 2009)) and dense retrievers (e.g. (Liu et al., 2021)). To further improve the effectiveness of the off-the-shelf retrievers, strategies for fine-tuning retrievers on specific target domains have been proposed such as PromptPG (Lu et al., 2022), UDR (Li et al., 2023c), and LLM-R (Wang et al., 2023), etc. Some works also adopt GPT to help retrieve and rerank samples by providing special prompts and related samples, such as Rerank (Sun et al., 2023b), SuRe (Kim et al., 2024), etc.

Long-tail knowledge learning for ICL. Kandpal et al. (2023) is the first to explore the influence of the long-tail distribution in pre-training data on LLM memorization. They find retrieval augmentation as a promising approach to significantly reduce the LLM's dependence on pre-training knowledge. Several subsequent works have built on this retrieval augmentation approach to address the long-tail problem in LLMs. For example, Dai et al. (2023) propose a retrieve-then-rerank framework leveraging knowledge distillation (KD) from the LLM to tackle long-tail QA. However, their method involves tuning the language model, which is computationally expensive and impractical for black-box LLMs such as GPT-4 (Achiam et al., 2023). Another line of research focuses on augmenting the training set using GPT (Saad-Falcon et al., 2023; Cloutier and Japkowicz, 2023; Li et al., 2023a), followed by fine-tuning the retriever to enhance its performance. Nonetheless, determining which samples should be augmented remains challenging. Augmenting the training set based on seed sentences often introduces repetitive rather than diverse information, and incurs significant costs due to GPT queries. Therefore, in this paper, rather than augmenting the training set for fine-tuning the retriever, we aim to train an effective retriever capable of selecting the most informative samples to augment the test query during inference.

3 Problem Formulation

In this paper, we target in-context learning (ICL) for QA tasks including multiple-choice QA and open-ended QA from different domains. Suppose we have a training set $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{N}$ related to the query domain, where x is the question and y is the answer. Given a query problem p_i from a test set $\mathcal{P}$ and a K-shot inference budget, we will retrieve K related samples $\mathcal{E}_i = \{e_i^k = (x_i, y_i) | e_i^k \in \mathcal{T}\}_{k=1}^{K}$ and construct a prompt $P(\mathcal{E}_i, p_i)$ as input

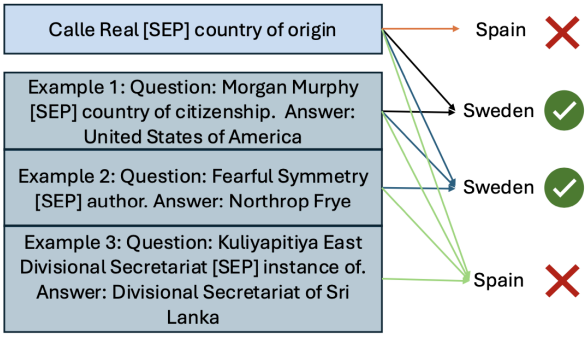


Figure 2: Case study for uncertainty of ICL.

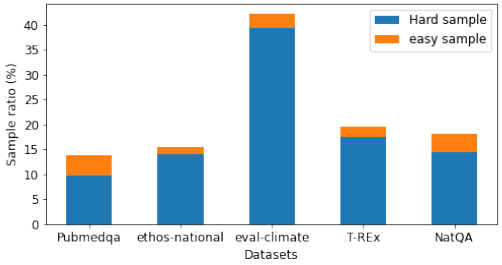


Figure 3: Uncertain sample ratios.

to feed into the LLM:

$$P(\mathcal{E}_i, p_i) = \pi(e_i^1) \oplus \cdots \oplus \pi(e_i^K) \oplus \pi(p_i, \cdot), \quad (1)$$

where π is the template for each sample. The predicted answer from the LLM for question p_i is given by:

$$\hat{a}_i = \text{LLM}(P(\mathcal{E}_i, p_i)). \quad (2)$$

4 Motivation: Uncertainty of In-context Learning

Due to the lack of knowledge of some specific domains during the pre-training stage, there exists long-tail knowledge that failed to be captured by the LLMs (Kandpal et al., 2023). We define easy samples as queries that have been captured during the LLM's pre-training stage and are stored in its memorization. In contrast, hard samples refer to queries that the LLM failed to capture, which are more likely to represent long-tail data. We classify easy and hard samples using the zero-shot testing results $\hat{a}_i = \text{LLM}_{\text{0-shot}}(p_i)$:

$$\begin{aligned} \mathcal{P}_{\text{easy}} &= \{(p_i, a_i) \in \mathcal{P} | \mathbb{1}(\hat{a}_i, a_i) = 1\}, \\ \mathcal{P}_{\text{hard}} &= \{(p_i, a_i) \in \mathcal{P} | \mathbb{1}(\hat{a}_i, a_i) = -1\}, \end{aligned} \quad (3)$$

where the indicator function $\mathbb{1}(\cdot)$ returns 1 if the predicted answer $\hat{a}_i$ aligns with the ground truth answer a_i, otherwise it returns -1. According to Kandpal et al. (2023), retrieval augmentation methods help alleviate the long-tail problem, as when a

retriever succeeds in finding the most relevant samples from the training set $\mathcal{T}$, it reduces the LLM's needs to have a large amount of related knowledge in its memorization. However, our experiments revealed that the LLMs exhibit higher uncertainty when presented with hard samples, regardless of the retrieval augmentation applied. Fig. 3 shows the uncertain sample ratios that experienced a prediction change on five datasets. Given a certain inference budget $K = 5$, 21.84% of queries experience a prediction change when we increase from 0-shot to 5-shot. Among these uncertain queries, 87.18% are hard samples and 12.82% samples are easy samples using BM25 retrieval (Robertson et al., 2009). For hard samples, even a tiny variation in retrieved set $\mathcal{E}$ can mislead the LLM's prediction. One case study for hard sample queries from T-REx (Elsahar et al., 2018) is shown in Fig. 2. In this case, LLM gives a correct answer with the first two informative samples in $\mathcal{E}$, effectively compensating for the LLM's long-tail knowledge. However, the answer gets wrong when a third sample is added to the prompt, which indicates the newly added knowledge is misleading. Other cases to show the uncertain prediction of LLM can be found in Fig. 7 in Section 6.4 and Table 6 in Appendix.

Given the uncertainty of in-context learning, our goal is to improve the prediction accuracy of hard samples while maintaining the prediction stability on easy samples. During testing, we lack prior knowledge to determine whether a query falls into the easy or hard category. The primary challenge, therefore, is to prevent the inclusion of misleading information in the retrieved set $\mathcal{E}$, which could lead to incorrect predictions. Simultaneously, we must ensure that the retrieved samples are sufficiently informative to address long-tail knowledge gaps and guide the LLM toward the correct answer.

5 In-context Learning with Dynamic Uncertainty Ranking

In this section, we introduce a dynamic uncertainty ranking method built on a reinforcement learning-based retriever. This method adjusts the retriever by applying a dynamic threshold, lowering the rankings of misleading samples while elevating the rankings of informative and stable ones.

5.1 Retrieved Sample Selection

The original training set $\mathcal{T}$ is randomly divided into a validation set $\mathcal{V}$, and a candidate pool $\mathcal{C}$, from which the retrieved sample set $\mathcal{E}$ is selected. Following Lu et al. (2022), the retriever structure is built upon BERT (Devlin, 2018) with a linear layer appended to the final pooling layer of the BERT model. During training, the BERT is frozen, and only the parameter $\theta = (\mathbf{W}, \mathbf{b})$ of the linear layer is fine-tuned. Given a query p_i from the validation set $\mathcal{V}$ and a retrieved sample e_i from $\mathcal{C}$, the ranking score of the retriever is achieved by the hidden logical similarity shared among samples:

$$S_\theta(e_i|p_i) = \frac{\exp[\mathbf{h}(e_i) \cdot \mathbf{h}(p_i)]}{\sum_{e_i' \in \mathcal{E}} \exp[\mathbf{h}(e_i') \cdot \mathbf{h}(p_i)]}, \quad (4)$$

where $\mathbf{h}(\cdot) = \mathbf{W}(\text{BERT}(\cdot)) + \mathbf{b}$ is the output of the linear layer.

To ensure the diversity and similarity of retrieved samples, and reduce the computational cost, we first adopt an off-the-shelf retriever BM25 (Robertson et al., 2009) to pre-select a small candidate set $\mathcal{C}_i'$ from the large candidate pool $\mathcal{C}$ following Rubin et al. (2021); Sun et al. (2023b); Kim et al. (2024).

Suppose the shot number is k, by selecting samples with the Top-k highest ranking score using our retriever S_θ, we can achieve the retrieved sample set $\mathcal{E}_i$ for p_i from candidate pool $\mathcal{C}_i'$ as follows:

$$\mathcal{E}_i = \{e_i^k \sim \text{Top-}k(S_\theta(e_i^k|p_i))|e_i^k \in \mathcal{C}_i'\}. \quad (5)$$

The retriever selection process for testing is the same as the training, the only difference is the validation set $\mathcal{V}$ will be replaced with the test set $\mathcal{P}$.

5.2 Retriever Training

Motivated by the exploration in Section 4, to improve retrieval augmentation for both hard and easy samples, we introduce a dynamic ranking method that updates the retriever using feedback from the LLM, driven by its varying responses to each retrieved sample.

Decide maximum shot number. Before training, we first decide the maximum shot number for each validation sample $p_i \in \mathcal{V}$. To achieve this, we define a maximum shot number budget K and a dynamic budget controller σ initialized as 0 for ranking scores S_θ. Only samples with ranking scores above the threshold σ will be selected to update the retriever. The maximum shot number k_i for p_i is:

$$k_i = \min(K, N_i^{\max}), \quad (6)$$

where $N_i^{\max} = |\{e_i^k \sim S_\theta(e_i^k|p_i)|e_i^k \in \mathcal{C}_i', S_\theta(e_i^k|p_i) > \sigma\}|$.

Training process. Given the maximum shot number k_i, we then conduct inference for p_i from 0-shot to k_i-shot to capture the effect of each retrieved sample on the LLM. The 0-shot inference on p_i can be considered as a means of long-tail sample detection as defined in Eq. (3). If the model's answer is incorrect, the sample is classified as a hard sample (i.e., long-tail sample), and the retrieved set should provide informative augmentation. Conversely, if the model produces the correct answer, the sample is classified as an easy sample, and the retrieved set should avoid introducing any misleading samples. We define the retrieved sample set for the j-shot inference as the top-j highest ranking score selected from candidate pool $\mathcal{C}_i'$:

$$\mathcal{E}_i^j = \{e_i^k \sim \text{Top-}j(S_\theta(e_i^k|p_i))|e_i^k \in \mathcal{C}_i'\}, \quad (7)$$
$$\text{where} \quad j = \{0, 1, \cdots, k_i\}.$$

The prediction from LLM based on $\mathcal{E}_i^j$ and p_i is generated according to Eq. (2) as $\hat{a}_i^j = \text{LLM}(P(\mathcal{E}_i^j, p_i))$. The retrieved sample's impact on the prediction is reflected by the reward function $R(\hat{a}_i^j, a_i) = \mathbb{1}(\hat{a}_i^j, a_i)$, where a_i is the ground truth answer for p_i, $\mathbb{1}(\cdot)$ is the indicator function.

Our training goal is to maximize the expected reward w.r.t. the parameters of the retriever using the Policy Gradient method (Sutton et al., 1999). Since the expected reward cannot be computed in closed form, following Lu et al. (2022), we compute an unbiased estimation with Monte Carlo Sampling:

$$\mathbb{E}_{e_i \sim S_\theta(e_i|p_i)}[(\hat{a}_i, a_i)] \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k_i} R(\hat{a}_i^j, a_i), \quad (8)$$

where N is the batch number yielded from $\mathcal{V}$. Following the REINFORCE policy gradient (Williams, 1992), we update the retriever using:

$$\begin{aligned}
&\nabla \mathbb{E}_{e_i \sim S_\theta(e_i|p_i)}[R(\hat{a}_i, a_i)] \\
=&\mathbb{E}_{e_i \sim S_\theta(e_i|p_i)} \nabla_\theta \log(S_\theta(e_i|p_i)) R(\hat{a}_i, a_i) \\
\approx&\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k_i} \nabla_\theta \log(S_\theta(e_i^j|p_i)) R(\hat{a}_i^j, a_i), \quad (9)
\end{aligned}$$

where $e_i^j = \mathcal{E}_i^j - \mathcal{E}_i^{j-1}$ is the difference between the retrieved sets for j-shot and $(j-1)$-shot. This approach incorporates the dynamic influence of each retrieved sample on the LLM, providing a better handling of uncertainty in ICL. Specifically, retrieved samples that yield correct predictions ($R(\cdot) = 1$) are treated as informative and contribute to augmenting long-tail knowledge, thus receiving a higher ranking. Conversely, retrieved samples that lead to incorrect predictions ($R(\cdot) = -1$) are considered misleading and are ranked lower.

Algorithm 1 ICL with dynamic uncertainty ranking

Input: Retriever S_θ, training set $\mathcal{T}$, maximum shot number K.
Output: Trained retriever S_θ .
Randomly split $\mathcal{T}$ into $\mathcal{V}$ and $\mathcal{C}$.
Initialize $\theta \leftarrow \theta_0$, threshold $\sigma \leftarrow 0$.
for $\mathcal{V}_{\text{batch}} \in \mathcal{V}$ **do**
  Initialize batch loss $L \leftarrow 0$.
  for each validation sample $p_i \in \mathcal{V}_{\text{batch}}$ **do**
    Pre-select $\mathcal{C}_i'$ from $\mathcal{C}$ using BM25 for p_i.
    Calculate the maximum shot number k_i based on σ using Eq. (6).
    for $j = 0, 1, \cdots, k_i$ **do**
      Get the retrieved set $\mathcal{E}_i^j$ using Eq. (7).
      Get prediction $\hat{a}_i^j = \text{LLM}(P(\mathcal{E}_i^j, p_i))$.
      Get reward $R(\hat{a}_i^j, a_i) = \mathbb{1}(\hat{a}_i^j, a_i)$.
      $L \leftarrow L - R(\hat{a}_i^j, a_i) \cdot \log(S_\theta(e_i^j|p_i))$.
      if $R(\hat{a}_i^j, a_i) = -1, R(\hat{a}_i^{j-1}, a_i) = 1$ **then**
        Update σ using Eq. (10).
      end if
    end for
  end for
  Optimize L w.r.t. θ using Eq. (9).
end for

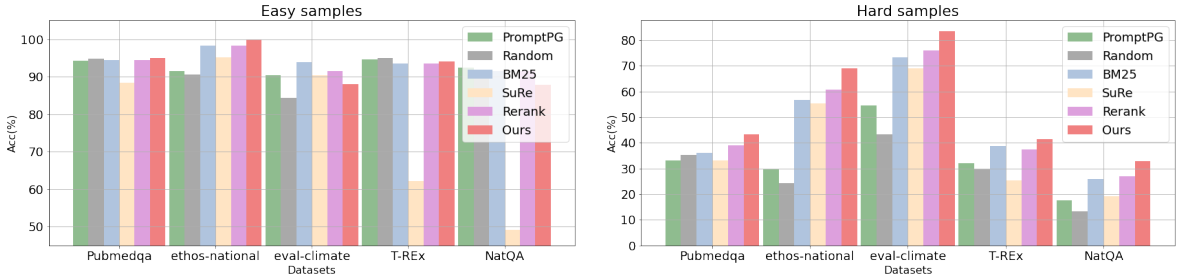
Update budget controller σ. In order to increase training efficiency and reduce the cost of querying the LLM, we also update the threshold σ that served as a budget controller at the turning point for prediction change to decrease the inference times while maintaining the effect of our training strategy. Specifically, we focus on a special case: when the LLM experiences a prediction change from true to false, i.e., $R(\hat{a}_i^{j-1}, a_i) = 1$ and $R(\hat{a}_i^j, a_i) = -1$. In this case, the first $(j-1)$-th samples have a positive impact on the inference of LLM, while the j-th sample has a negative impact. Thus, we update the threshold σ as the maximum value of the ranking score for unselected samples in $\mathcal{E}_i^{k_i}$ for the $(j-1)$-shot round as follows:

$$\sigma = \max(S_\theta(e_i^k|p_i)), \quad e_i^k \in \mathcal{E}_i^{k_i} - \mathcal{E}_i^{j-1}. \quad (10)$$

Since we only select samples with ranking scores larger than σ as shown in Eq. (6), the retrieved sam-

Retrieval Method	Dataset					Avg
	Pubmedqa	ethos-national	eval-climate	T-REx	NatQA	
0-shot	72.87 ± 0.31	75.61 ± 0.51	46.30 ± 0.32	42.60 ± 2.36	44.20 ± 1.91	56.32 ± 1.08
Random sampling	78.20 ± 0.53	75.17 ± 1.01	66.30 ± 3.53	57.13 ± 1.97	46.80 ± 1.44	64.72 ± 1.70
BM25	78.93 ± 0.31	87.47 ± 0.39	82.57 ± 0.30	62.13 ± 1.33	55.00 ± 1.14	73.22 ± 0.69
SuRe	78.93 ± 0.42	85.23 ± 0.33	78.89 ± 0.30	39.80 ± 0.57	32.00 ± 3.40	62.97 ± 1.00
Rerank	78.93 ± 0.42	89.15 ± 0.39	83.22 ± 0.32	62.07 ± 2.01	53.80 ± 1.91	73.43 ± 1.01
PromptPG	78.47 ± 0.90	77.74 ± 2.16	72.78 ± 2.00	60.73 ± 3.21	50.80 ± 2.00	68.10 ± 2.05
Ours	$\mathbf{80.60 \pm 0.35}$	$\mathbf{92.40 \pm 0.20}$	$\mathbf{85.37 \pm 0.32}$	$\mathbf{65.00 \pm 2.69}$	$\mathbf{57.60 \pm 1.91}$	$\mathbf{76.19 \pm 1.09}$

Table 1: Comparison results between proposed methods and baselines on QA tasks from different domains.



(a) Easy sample accuracy.

(b) Hard sample accuracy.

Figure 4: Accuracy on easy and hard samples for proposed method and baselines.

ples that serve as a good compensation for long-tail knowledge will be ranked higher, and be used for updating the retriever more frequently. Note that updating σ will not wipe out the updating of misleading samples, as the turning point for prediction change is different for each validation sample. Without affecting our original training strategy, we improve the efficiency and deduct the querying cost. Our algorithm is summarized in Algorithm 1.

6 Experiments

In this section, we first introduce the experiment setup and then show the effectiveness of our method through various empirical results.

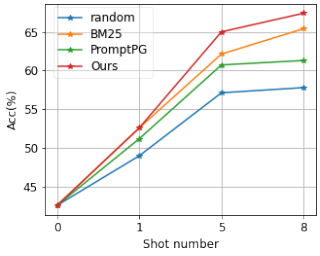
6.1 Experimental Setup

Datasets: We conduct the experiments on QA datasets from different domains, including three multi-choice datasets: biomedical dataset *Pubmedqa* (Jin et al., 2019), speech detection dataset *ethos-national* (Mollas et al., 2022), climate change dataset *eval-climate* (Barbieri et al., 2020), and two open-ended QA dataset: *T-REx* (Elsahar et al., 2018) and NaturalQuestions (*NatQA*) (Kwiatkowski et al., 2019). More datasets details can be found in Appendix A.1.

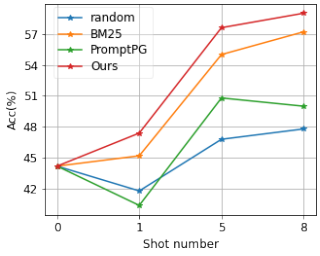
Baselines: We compare our method with six baselines, including *0-shot* inference and five few-shot retrieval augmentation methods. The retrieval augmentation methods are as follows: 1) *Random sampling*: selecting ICL samples from the candidate set, a widely adopted practice in many ICL studies (Wei et al., 2022; Wang et al., 2022); 2) *BM25* (Robertson et al., 2009): an off-the-shelf sparse retriever; 3) SuRe (Kim et al., 2024): first use GPT to summarize the retrieved passages from BM25 for multiple answer candidates, then determines the most plausible answer by evaluating and ranking the generated summaries; 4) Rerank (Sun et al., 2023b): use GPT to rerank samples retrieved by BM25; 5) *PromptPG* (Lu et al., 2022): a BERT-based dense retriever trained using reinforcement learning based on the feedback from GPT.

Evaluation: For multi-choice QA, we use accuracy for evaluation. For open-ended QA, we use Normalized Exact Match (NEM), which evaluates whether the normalized string output by the inference LLM is identical to the reference string.

Implementation: The LLM used in our experiment is GPT-4 (Achiam et al., 2023). Due to the limited data size in *tweet_eval-stance_climate*, the training set is split into 50 candidate samples and 150 validation samples. For the other datasets, we use 1000 samples in the candidate pool and 200 samples in the validation set. All methods share the same train-test split. The number of pre-selected samples in $\mathcal{C}'$ is set to 20 by default for both the training and testing stages. For the few-shot case, the shot number is set to 5, unless otherwise specified. During the training of our method, the maximum shot number budget K is also set to 5. The batch size is set to 20. Experiments for all test datasets are repeated 3 times with different seeds, and the average accuracy is reported in the results.



(a) T-REx.



(b) NatQA.

Figure 5: Effects of different number of shots.

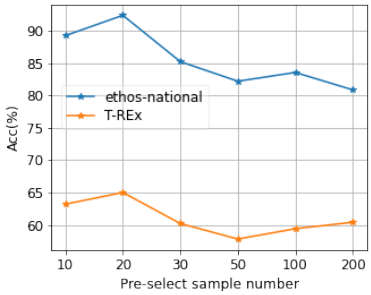


Figure 6: Effects of different pre-select numbers.

6.2 Main Results

Table 1 presents the mean and standard deviation (std) of accuracy for our proposed method and the baselines across five QA datasets. Our approach outperformed all baselines across tasks, with an average improvement of 2.97% ranging from 1.67% to 3.25% over the best baseline. The trained retriever PromptPG gives the most uncertain prediction with a std of 2.05%. Although our method is based on PromptPG, by giving informative and stable samples higher ranks, we not only improve the overall accuracy but also decrease std to 1.09%, comparable to 0-shot inference.

We further investigate the accuracy of easy and hard samples in Fig. 4. As illustrated in Eq. (3), the easy/hard sample classification is decided by the 0-shot inference results, and the hard samples can be considered as long-tail questions of GPT-4. First, we observe a similar pattern to Kandpal et al. (2023) that retrieval augmentation greatly improves the accuracy of long-tail samples. This could come from various aspects of augmented samples—such as label space, input text distribution, and sequence format—that collectively improve final predictions (Min et al., 2022). Compared with 0-shot inference, even random sampling improves accuracy on hard samples from 0% to 29.17%. However, retrieval augmentation is highly dependent on the quality of the retrieval set. By retrieving the most similar samples, BM25 achieves an accuracy of 46.12%. Rerank further improves the accuracy to 48.03%. Our method includes the most informative samples based on the sample-wise feedback from LLM, and improves the accuracy on hard samples to 53.99%, which surpasses the best baseline with a large average margin of 5.96% ranging from 2.69% to 8.11%, while maintaining the accuracy on easy samples.

Dataset	PromptPG	UR	PromptPG+PS	UR+PS (Ours)
ethos-national	77.74	81.21	86.91	**92.40**
Pubmedqa	78.47	80.10	79.10	**80.60**

Table 2: Effects of different components. PS denotes pre-selection. UR denotes uncertainty rank.

6.3 Ablation Studies

Effects of different components. We verify the effectiveness of two components of our proposed method: uncertainty rank and pre-selection in Table 2. We first compared the uncertainty rank (UR) strategy with another trained retriever PromptPG which shared the same retriever architecture as ours. We improve the accuracy by 3.47% and 1.63% for two different datasets. PromptPG adjusts the ranking of candidate samples based on the feedback on the entire retrieved set for the validation samples, while UR raises the ranks for informative and stable samples and lowers the ranks for misleading samples based on the sample-wise feedback from LLMs. UR avoids the condition when misleading samples are included and negatively changes the answer from true to false. In this way, UR greatly enhances the retrieved sample set for augmentation.

The second component pre-selection (PS) improves the results of both PromptPG and UR by selecting more diverse and similar related samples in the candidate set $\mathcal{C}'$. Then the second step retrieval can select samples from a smaller candidate pool of higher quality. By combining these two components together, we can achieve an overall improvement of 14.66% and 2.13% for two different datasets. The improvement on ethos-national is more significant than Pubmedqa because the predicted answer on ethos-national is more uncertain given different combinations of retrieved samples.

Effects of different number of shots. We show the effects of different shot numbers for two datasets in Fig. 5 where our method consistently

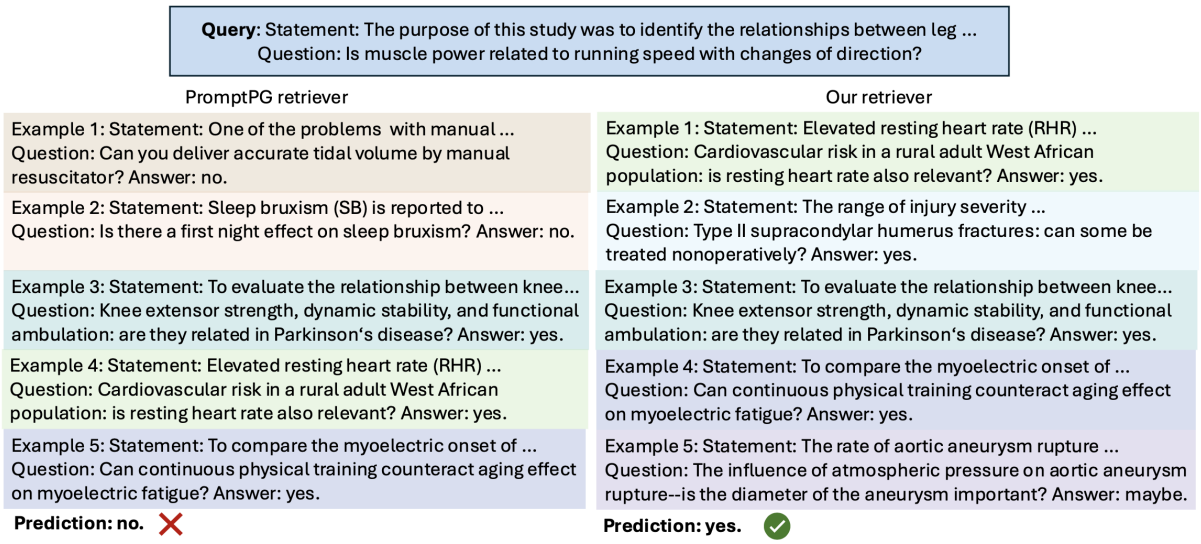


Figure 7: Case study for retrieved samples of hard samples.

outperforms other baselines. For NatQA, the accuracy of random sampling and PromptPG retrieval does not monotonically increase with shot number due to low-quality, misleading samples, which can degrade performance. In contrast, our method prioritizes high-quality samples, and as the number of shots increases, the advantages of our algorithm become more pronounced, resulting in improved accuracy.

Effects of different number of pre-selection samples. In Fig. 6, we investigate how the number of pre-seletion samples impacts our algorithm. For both datasets, the accuracy first increases and then decreases. If too few samples are selected, the candidate pool $\mathcal{C}'$ for our reinforcement learning -based ranking stage lacks diversity, limiting the policy gradient strategy's action space. Consequently, the learned retriever struggles to find the most informative samples. If the number is too large, $\mathcal{C}'$ includes many irrelevant samples, making it difficult for the policy gradient strategy to learn an optimal solution in the large search space (Lu et al., 2022). This can lead the retriever to capture irrelevant or misleading information.

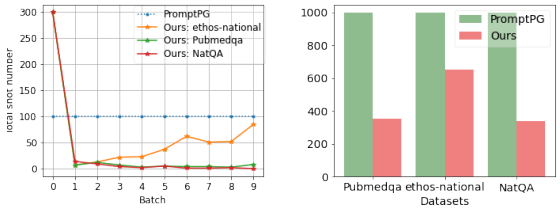
6.4 Case Study

To intuitively show the effectiveness of our proposed method on hard samples, we show one case on Pubmedqa by comparing the retrieved samples of PromptPG retriever and our retriever in Fig. 7. According to this case, the two retrieved sets even have three overlap samples (marked as the same color), but the prediction is completely different. PromptPG gives a wrong prediction answer, while our method delivers the right answer. This result verifies that GPT-4 gives uncertain predictions on long-tail samples. Since 0-shot inference gives a wrong prediction answer on this query question, the informative augmented information can be contained in the retrieved set of our method (see right column), while for PromptPG, misleading information can be contained in the two samples that do not intersect with our retriever set (see left column), which shifts the predicted answer from true to false. Compared with PromptPG, our retriever ranks the three overlapped samples higher and gives two more informative samples. With the combination effect of these two, our method gives the correct prediction. More cases on hard samples from other datasets can be found in Table 7 in the appendix.

6.5 Efficiency Analysis

Query cost. We set threshold σ as the budget controller to reduce the cost of the querying GPT-4. Since the query cost depends on token length, we compare the query costs of our method and PromptPG (both trained based on GPT-4) in Fig. 8a. Specifically, we calculate the total number of shots included in each query during training for each batch within one epoch for both methods. The blue dash line shows the total shot number of PromptPG for all datasets, since the batch size is 20, and the shot number is fixed at 5, the total shot number is fixed at 100 for each batch. According to the results, only batch 0 of our method surpasses PromptPG with a total shot count of 300. For subsequent batches, as the threshold σ is adjusted based on changes in the LLM's pre-

dictions, the query shot count drops significantly, resulting in the total shot count consistently being lower than that of PromptPG. Aggregating the shot numbers across 10 batches, our method achieves only 33.8%, 65.2%, and 35.3% of the shot count of PromptPG on Pubmedqa, ethos-national, and NatQA, respectively as shown in Fig. 8b. Thus, in conjunction with the accuracy comparison presented in Table 1, our approach not only enhances query accuracy but also reduces the overall query cost.



(a) Shot # w.r.t. the batch. (b) Shot # of one epoch.

Figure 8: Efficiency analysis.

Convergence speed. We empirically demonstrate the convergence speed by showing training loss curves in Fig. 9. According to the results, the training loss quickly converges to a small value close to 0 within 15 batchs, which verify the high computational efficiency of our method.

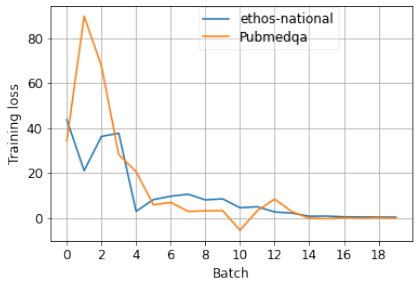


Figure 9: Training loss w.r.t. batch.

6.6 Transferability Analysis

We investigate the transferability of our retriever in Table 3. We use our retriever trained on dataset *ethos-national*, and evaluate its cross-domain effectiveness across the rest of the four datasets. Although the cross-domain results are still slightly inferior to the in-domain results, the performance gap is minimal, averaging only 0.98%. Furthermore, the cross-domain results outperform the best baseline. These findings indicate that our trained ranking strategy is transferable to other datasets, providing a cost-effective alternative to retraining.

	Pubmedqa	eval-climate	NatQA	T-REx	Avg
Best baseline	78.93	83.22	55.00	62.13	69.82
Ours: cross-domain	79.60	83.33	57.20	64.50	71.16
Ours: in-domain	80.60	85.37	57.60	65.00	72.14

Table 3: Transferability of our method.

7 Conclusion

In this paper, to improve the uncertain prediction of LLMs on long-tail knowledge, we propose a reinforcement learning-based dynamic uncertainty ranking method for retrieval-augmented ICL with a budget controller. Specifically, it considers the dynamic impact of each retrieved sample based on the LLM's feedback. Our ranking system system raises the ranks of more informative and stable samples and lower the ranks of misleading samples efficiently. Evaluations of various QA datasets from different domains show that our proposed method outperformed all the baselines, and especially improve the LLM's prediction on the long-tail questions.

8 Limitations

There are several limitations of our work.

First, our method do not consider the effect of different orders within the retrieved set and rank the retrieved samples according to their ranking scores. Future works can be extended based on our work by considering different inner order within the retrieved set and their effect on the prediction results.

Second, although our experimental results show that our method greatly improves the prediction accuracy on long-tail samples, our method cannot handle query cases with no related knowledge either in the pre-training set or candidate pool.

Third, our method focused on QA tasks using LLM. For future work, our method can be extended to other tasks such as summarization, translation, and recommendation as follows. Since our method is to train a reranker based on the reward signal from LLM, to adapt to other tasks, we can modify the evaluation score that is used to determine the reward. If the accuracy of the LLM's predicted answer is unavailable, alternative metrics such as BLEU and ROUGE can be used to assess the consistency between the prediction and the ground truth. A threshold can then be set for these scores, where values exceeding the threshold yield a positive reward, while lower values result in a negative reward.

References

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

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Francesco Barbieri, Jose Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. 2020. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. *arXiv preprint arXiv:2010.12421*.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Jiuhai Chen, Lichang Chen, Chen Zhu, and Tianyi Zhou. 2023. How many demonstrations do you need for in-context learning? *arXiv preprint arXiv:2303.08119*.

Nicolas Antonio Cloutier and Nathalie Japkowicz. 2023. Fine-tuned generative llm oversampling can improve performance over traditional techniques on multiclass imbalanced text classification. In *2023 IEEE International Conference on Big Data (BigData)*, pages 5181–5186. IEEE.

Yi Dai, Hao Lang, Yinhe Zheng, Fei Huang, and Yongbin Li. 2023. Long-tailed question answering in an open world. *arXiv preprint arXiv:2305.06557*.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR.

Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. 2024. Sure: Summarizing retrievals using answer candidates for open-domain qa of llms. *arXiv preprint arXiv:2404.13081*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Huihan Li, Yuting Ning, Zeyi Liao, Siyuan Wang, Xiang Lorraine Li, Ximing Lu, Faeze Brahman, Wenting Zhao, Yejin Choi, and Xiang Ren. 2023a. In search of the long-tail: Systematic generation of long-tail knowledge via logical rule guided search. *arXiv preprint arXiv:2311.07237*.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. 2023b. Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*.

Xiang Li, Haoran Tang, Siyu Chen, Ziwei Wang, Ryan Chen, and Marcin Abram. 2024. Why does in-context learning fail sometimes? evaluating in-context learning on open and closed questions. *arXiv preprint arXiv:2407.02028*.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023c. Unified demonstration retriever for in-context learning. *arXiv preprint arXiv:2305.04320*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. 2019. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2537–2546.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2022. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

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*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2022. Ethos: a multi-label hate speech detection dataset. *Complex & Intelligent Systems*, 8(6):4663–4678.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Jon Saad-Falcon, Omar Khattab, Keshav Santhanam, Radu Florian, Martin Franz, Salim Roukos, Avirup Sil, Md Arafat Sultan, and Christopher Potts. 2023. Udapdr: unsupervised domain adaptation via llm prompting and distillation of rerankers. *arXiv preprint arXiv:2303.00807*.

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2023a. Head-to-tail: How knowledgeable are large language models (llm). *AKA will llms replace knowledge graphs*.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023b. Is chatgpt good at search? investigating large language models as re-ranking agents. *arXiv preprint arXiv:2304.09542*.

Richard S Sutton, Andrew G Barto, et al. 1999. Reinforcement learning. *Journal of Cognitive Neuroscience*, 11(1):126–134.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Dave Van Veen, Cara Van Uden, Louis Blankemeier, Jean-Benoit Delbrouck, Asad Aali, Christian Bluethgen, Anuj Pareek, Malgorzata Polacin, Eduardo Pontes Reis, Anna Seehofnerová, et al. 2024. Adapted large language models can outperform medical experts in clinical text summarization. *Nature medicine*, 30(4):1134–1142.

Fali Wang, Runxue Bao, Suhang Wang, Wenchao Yu, Yanchi Liu, Wei Cheng, and Haifeng Chen. 2024a. Infuserki: Enhancing large language models with knowledge graphs via infuser-guided knowledge integration. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3675–3688.

Liang Wang, Nan Yang, and Furu Wei. 2023. Learning to retrieve in-context examples for large language models. *arXiv preprint arXiv:2307.07164*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Zhepeng Wang, Runxue Bao, Yawen Wu, Jackson Taylor, Cao Xiao, Feng Zheng, Weiwen Jiang, Shangqian Gao, and Yanfu Zhang. 2024b. Unlocking memorization in large language models with dynamic soft prompting. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9782–9796.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.

Haoyu Zhang, Jianjun Xu, and Ji Wang. 2019. Pretraining-based natural language generation for text summarization. *arXiv preprint arXiv:1902.09243*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.

A Appendix

A.1 Experiment Details

Dataset Details. In this paper, we evaluate across five QA datasets from different domains including multi-choice QA and open-ended QA. The detailed statistics of these datasets and the prompt format we used are shown in Table 4 and Table 5. We conduct the train-test split for the last four datasets following Li et al. (2024). We randomly sample 1000 samples from the training dataset if the training set size exceeds 1000 to simulate the scenario where only a limited number of samples can be collected.

A.2 Extended Experimental Results

More case study on the uncertainty of ICL. According to Table 6 on the healthcare dataset Pubmedqa, LLM can achieve correct prediction with the first two retrieved samples but gives a wrong prediction when the third sample is added to the prompt, which indicates that the third sample is misleading.

More case study on hard samples. Table 7 shows another case for retrieved samples of hard samples on T-REx. According to the results, the query question asks about the instance of a subject, while the prompt retriever retrieved samples about the questions related to the locations, which mislead the final prediction. For our retriever, all the retrieved samples are related to the questions related to the instance of the subject, and provide informative augmentations for the inference.

Dataset	Type	Domain	Training	Test	Prompt format
Pubmedqa	Multi-choice	Healthcare	1000	500	SQO-A
ethos-national	Multi-choice	Speech detection	476	298	QO-A
eval-climate	Multi-choice	climate change	288	180	QO-A
T-REx	Open-ended	Wikipedia	20128	5032	Q-A
NatQA	Open-ended	Wikipedia	11476	2869	Q-A

Table 4: The statistics of the datasets used in this paper.

Notation	Retrieval sample format	Query sample format
Q-A	Question: <question> Answer: The answer is <answer>	Question: <question> Answer:
QO-A	Question: <question> Options: (A) <option A> (B) <option B> (C) <options C>... Answer: The answer is <answer>	Question: <question> Options: (A) <option A> (B) <option B> (C) <options C>... Answer:
SQO-A	Statement: <context> Question: <question> Options: (A) <option A> (B) <option B> (C) <options C>... Answer: The answer is <answer>	Statement: <context> Question: <question> Options: (A) <option A> (B) <option B> (C) <options C>... Answer:

Table 5: Prompt Format notations.

Query: Statement: Lymphedema may be identified by... Question: Can a practicing surgeon detect early lymphedema reliably?		
	Retrieved samples	Prediction
Retrieved sample 1	Statement: Minority patients with cancer experience... Question: Can patient coaching reduce racial/ethnic disparities in cancer pain control? Answer: Yes.	Maybe (✓)
Retrieved sample 1 + sample 2	Statement: Minority patients with cancer experience... Question: Can patient coaching reduce racial/ethnic disparities in cancer pain control? Answer: Yes. Statement: The potential effects of binge drinking during pregnancy... Question: Does binge drinking during early pregnancy increase the risk of psychomotor deficits? Answer: No.	Maybe (✓)
Retrieved sample 1 + sample 2 + sample 3	Statement: Minority patients with cancer experience... Question: Can patient coaching reduce racial/ethnic disparities in cancer pain control? Answer: Yes. Statement: The potential effects of binge drinking during pregnancy... Question: Does binge drinking during early pregnancy increase the risk of psychomotor deficits? Answer: No. Statement: Despite the advantages from using aromatase inhibitors... Question: Do adjuvant aromatase inhibitors increase the cardiovascular risk in postmenopausal women with early breast cancer? Answer: Yes.	No (✗)

Table 6: Extended case study for the uncertainty of ICL on Pubmedqa.

Query question	Outlaw [SEP] instance of.	
Retriever	PromptPG retriever	Our retriever
Retrieved samples	Question: Hingani Dam [SEP] country. Answer: The answer is India.	Question: Schleich [SEP] instance of. Answer: The answer is municipality of Germany.
	Question: Maryland State Archives [SEP] applies to jurisdiction. Answer: The answer is Maryland.	Question: Chevry-sous-le-Bignon [SEP] instance of. Answer: The answer is commune of France.
	Question: Silvia Panguana [SEP] country of citizenship. Answer: The answer is Mozambique.	Question: The Listel Hotel [SEP] instance of. Answer: The answer is hotel.
	Question: New Paluvayi [SEP] located in the administrative territorial entity. Answer: The answer is Andhra Pradesh.	Question: Westona [SEP] instance of. Answer: The answer is railway station.
	Question: The '59 Sound [SEP] country of origin. Answer: The answer is United States of America.	Question: Secu [SEP] instance of. Answer: The answer is commune of Romania.
Prediction	film. (✗)	wooden roller coaster. (✓)

Table 7: Extended case study for retrieved samples of hard samples on T-REx.