Making RALM Robust to Irrelevant Contexts via Layer Knowledge Guided Attention

Weijie Shi[1*†], Hao Chen[2*], Jiaming Li[2], Yao Zhao[2*],
Yazhong Zhang[2‡], Qijin Chen[2‡], Jipeng Zhang[1], Ruiyuan Zhang[1],
Jia Zhu[3‡], Jiajie Xu[4‡], Xiaofang Zhou[1]

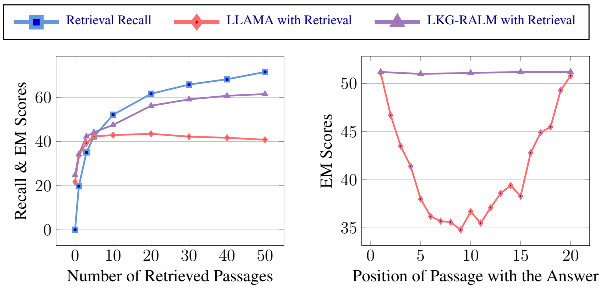
[1]The Hong Kong University of Science and Technology, [2]Alibaba Group,
[3]Zhejiang Key Laboratory of Intelligent Education Technology and Application,
Zhejiang Normal University, [4]Soochow University

Abstract

Retrieval-augmented language models (RALMs) aim to incorporate external knowledge to address the issues of factual hallucination and knowledge obsolescence faced by large language models (LLMs). Inevitably, the retrieved passages based on similarity search may be irrelevant to the given question, and the aggregation of these passages can confuse the model to give a correct answer. To improve the performance of RALM in such conditions, we propose layer-knowledge guided attention for RALMs, which harnesses the layer-wise knowledge of LLMs to optimize per-layer attention on useful passages, making the model pay attention to the most relevant content and ignore irrelevant ones. Specifically, we first systematically study LLM's attention patterns and their relationship with the accuracy of RALM responses, where middle-focus attentions play a crucial role in selectively gathering relevant information. Based on this, a layer-wise passage estimator leverages the varied knowledge encoded across LLM layers to assess not only passage relevance scores but also associated confidences. Finally, a relevance-aware passage fusion enables selective attention to relevant passages, mitigating distractibility and positional bias of causal attention. Experiments show that our method outperforms existing methods on RALM benchmarks.

1 Introduction

Large language models (LLMs) have demonstrated remarkable performance, scalability, and adaptability in various natural language processing tasks (Bang et al., 2023; Guo et al., 2023; Chowdhery et al., 2022). However, LLMs encounter significant challenges when tackling knowledge-intensive tasks, including factual hallucination (Cao et al., 2020; Raunak et al., 2021; Ji et al., 2023), knowledge obsolescence (He et al., 2022), and a lack of domain-specific expertise (Shen et al., 2023; Li et al., 2023). To address these issues, retrieval-augmented language model (RALM) has emerged as a mainstream approach, which leverages a retrieval-then-read pipeline to supply external information for the LLM answering questions.



(a) Accuracy plateaus due to *attention's distractibility*.

(b) U-shaped performance due to *positional bias*.

Figure 1: Comparison of LLAMA's and LKG-RALM's performance with retrieved passages.

Despite RALM's potential, LLMs struggle to handle retrieved passages, which contain irrelevant ones, hindering performance in two aspects:

- **Attention Distractibility**: As shown in Figure 1(a), while increasing retrieved passages improves recall linearly, LLM accuracy plateaus or declines due to attention disruption from irrelevant content (Shi et al., 2023a). The question tokens' attention becomes scattered across noisy information in the passages.

- **Positional Bias**: As illustrated in Figure 1(b), LLM performance exhibits a U-shaped curve based on passage position, with better handling of information at the start and end while missing crucial middle content (Liu et al., 2024a). This stems from LLM attention's

[*]Co-authors: Hao Chen, Yao Zhao

[†] **Email:** wshiah@connect.ust.hk

[‡]Corresponding authors: Yazhong Zhang, Qijin Chen, Jiajie Xu, Jia Zhu

Findings of the Association for Computational Linguistics: ACL 2025, pages 3652–3668
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

over-reliance on positional information.

While RankRAG (Yoran et al., 2024) attempts to address these issues by filtering irrelevant passages and optimizing passage placement, these serve as compromised strategies rather than fundamental fixes to LLM attention processing retrieved passages.

In this paper, we propose Layer-Knowledge Guided Attention for RALM (LKG-RALM), which harnesses the layer-wise knowledge of LLMs to optimize attention on useful passages. To effectively guide LLM's attention, accurately assessing the relevance of retrieved passages is crucial. Recent works demonstrate that LLM-based embeddings significantly outperform BERT-like models on the MTEB leaderboard due to superior scaling and comprehensive pre-training. Furthermore, Meng et al. (2022); Chuang et al. (2023) indicate that different LLM layers encode varied knowledge, from grammatical understanding in lower layers to reasoning capabilities in higher ones. Building on these insights, we propose a layer-wise passage estimator, which fully leverages varied knowledge of LLM layers to accurately predict both relevance and estimation confidence. Since not all layers' knowledge contributes equally to relevance assessment, an entropy-based layer-knowledge selection is proposed to dynamically determine which layers' knowledge is suitable for passages. To mitigate distractibility and positional bias from irrelevant passages, a relevance-aware passage fusion employs a relevance-guided attention mask to enable question tokens to selectively attend to retrieved passages for middle-focused attention patterns. Experiments demonstrate that LKG-RALM achieves substantial performance improvements across RALM datasets. Our contributions are summarized as:

- We present the first systematic study on the relation between RALM's attention patterns and performance. Based on these, LKG-RALM leverages layer-wise knowledge to guide middle-focused attention toward relevant passages, thereby enhancing the understanding of retrieved information.

- We propose a layer-wise passage estimator to utilize LLM layer-specific knowledge to assess reliable and adaptable passage relevance.

- We propose relevance-aware passage fusion to enable question tokens to selectively attend to relevant passages, mitigating distractibility and positional bias.

2 Related Work

2.1 Retrieval-augmented Language Model

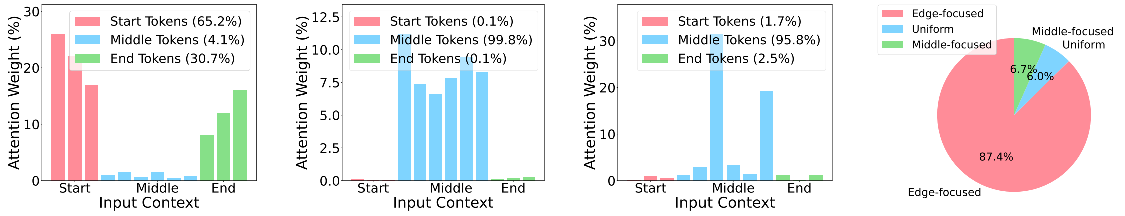
Retrieval-augmented language models (RALMs) (Zhao et al., 2024, 2023; Gao et al., 2023) enhance generation by incorporating retrieved passages through three main approaches: query-based fusion, which concatenates passages with input queries (Shi et al., 2023b; Ram et al., 2023) or features (Izacard and Grave, 2020; Liu et al., 2023); logits-based fusion, which combines probability distributions from input and retrieved passages (Khandelwal et al., 2019; Huang et al., 2023); and latent fusion, which integrates passages into hidden states via attention (Wang et al., 2023a) or weighted additions (Wu et al., 2024a).

Recent work has focused on addressing noise in retrieved passages. Liu et al. (2024a) analyzed position bias across model types and query positions, while Shi et al. (2023a); Wu et al. (2024b) attempted to incorporate passage relevance into context. Other approaches include filtering irrelevant passages (Zhang et al., 2021; Yoran et al., 2024) and developing noise-resistant fine-tuning strategies (Liu et al., 2024c; Yu et al., 2024). However, these methods remain constrained by reranking accuracy and fail to address the fundamental limitations of causal attention. Our work investigates the relationship between attention patterns and RALM performance, leading to our LKG-RALM approach that leverages layer-wise knowledge for improved passage attention.

2.2 Passage Relevance Assessment

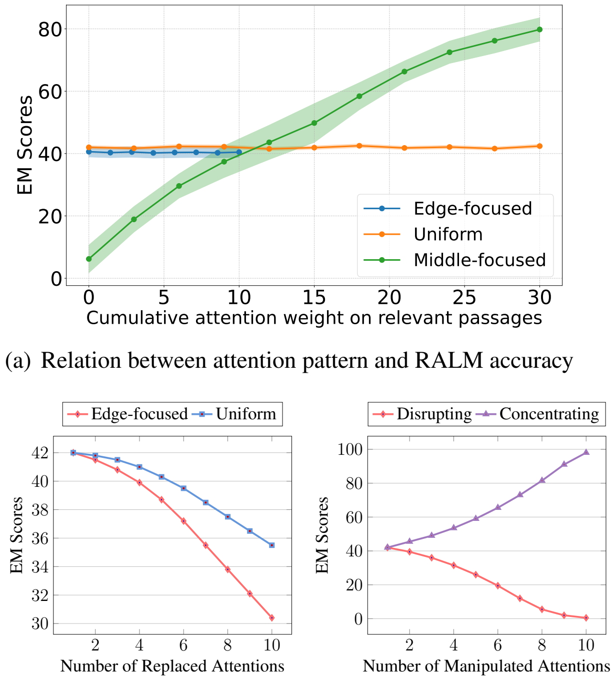
While traditional methods like BM25 (Robertson et al., 2009) and BERT-based models (Karpukhin et al., 2020; Izacard et al., 2021; Chen et al., 2024) have advanced text representation, they face scaling challenges in representation training. Recent approaches (Wang et al., 2023b; BehnamGhader et al., 2024; Springer et al., 2024) have shown promise in adapting decoder-only LLMs as text encoders through contrastive learning. However, even state-of-the-art models achieve only 62% accuracy on the MTEB leaderboard (Muennighoff et al., 2022), highlighting the need for more nuanced relevance assessment approaches.

Meng et al. (2022); Chuang et al. (2023); Zhang et al. (2024) have shown that LLMs encode layer-



(a) Edge-focused Attention (b) Uniform Attention (c) Middle-focused Attention (d) Attention Distribution

Figure 2: Attention patterns of RALM.



(a) Relation between attention pattern and RALM accuracy

(b) Impact of replacing edge-focused and uniform attention with middle-focused attention on relevant passages

(c) Impact of manipulating middle-focused attention patterns on relevant passages

Figure 3: Impact of manipulating attention patterns on RALM performance.

specific knowledge, ranging from grammatical structures in lower layers to complex reasoning in higher ones. Building on this insight, we propose a layer-wise passage estimator that leverages this hierarchical knowledge structure to provide comprehensive relevance assessments with reliability measures.

3 Preliminaries

3.1 Problem Formalization

Our method is depicted under the open question-answering (open-QA) settings, aiming to predict an answer y_{ans} based on a question q and n retrieved passages $[p_1, \ldots, p_n]$.

3.2 Analysis of Attention Patterns of RALM

To address the challenges of attention's distractibility and positional bias in RALM, it is crucial to systematically investigate its attention patterns. The attention mechanism selects specific tokens to gather information from retrieved passages for the generation of next token. Following the Fu (2024) methodology, we conducted a systematic study on the attention distribution of LLAMA-3.1-8B using 2000 samples from the NQ and TriviaQA dataset (details in Appendix A). Figure 2 reveals three distinct attention patterns that potentially impact the model's ability to process retrieved passages.

Edge-focused attention, observed in 78% of attention heads, shows over 99% of attention concentrating on the beginning and end of the context. Xiao et al. (2023) demonstrated that this phenomenon persists even when replacing the initial tokens with meaningless ones, indicating that the model emphasizes absolute position rather than semantic value. This pattern correlates strongly with positional bias, hindering the model's ability to process crucial information in the middle of the input sequence.

Uniform attention, accounting for 5.37% of patterns, distributes attention almost uniformly across all tokens in the context. While appearing to provide equal consideration to all information, this pattern potentially contributes to the model's distractibility by failing to focus on the most relevant parts of the input.

Middle-focused attention, though present in only 6% of attention heads, manifests in two variants: "scattered over middle" and "concentrated on middle". The former distributes attention across several tokens, while the latter concentrates on only one or two tokens. This pattern plays a crucial role in selectively gathering information from the context, essential for comprehending retrieved passages.

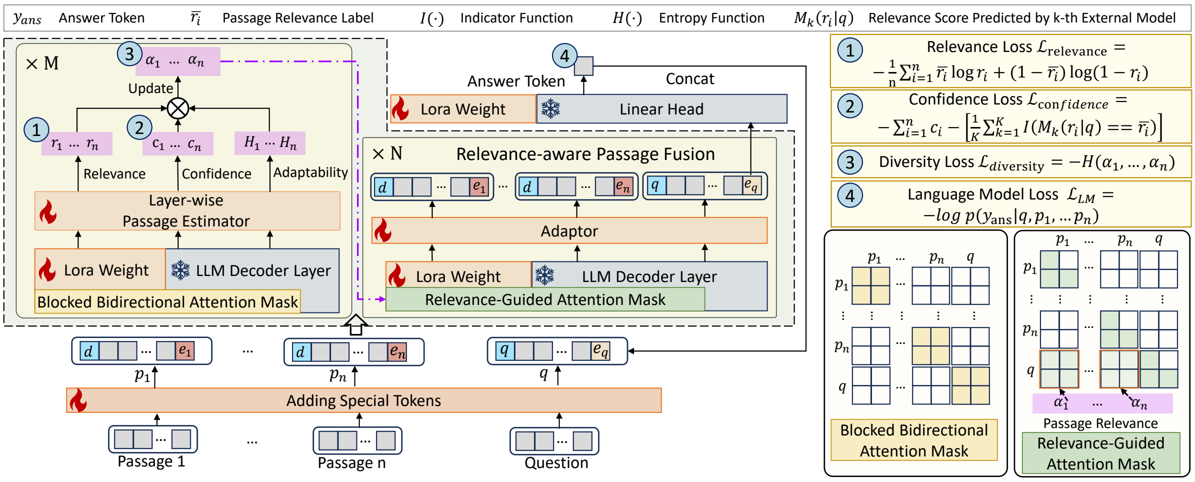

y_{ans} Answer Token
$\overline{r_i}$ Passage Relevance Label
$I(\cdot)$ Indicator Function
$H(\cdot)$ Entropy Function
$M_k(r_i|q)$ Relevance Score Predicted by k-th External Model
× M
3 $\alpha_1 \cdots \alpha_n$
Update
1 $r_1 \cdots r_n$
2 $c_1 \cdots c_n$
$H_1 \cdots H_n$
Relevance
Confidence
Adaptability
Layer-wise Passage Estimator
Lora Weight
LLM Decoder Layer
Blocked Bidirectional Attention Mask
4
Answer Token
Concat
Lora Weight
Linear Head
× N
Relevance-aware Passage Fusion
d … e_1
d … e_n
q … e_q
Adaptor
Lora Weight
LLM Decoder Layer
Relevance-Guided Attention Mask
p_1
p_n
q
Adding Special Tokens
Passage 1
Passage n
Question
1 Relevance Loss $\mathcal{L}_{relevance}$ = $-\frac{1}{n}\sum_{i=1}^{n}\overline{r_i}\log r_i + (1-\overline{r_i})\log(1-r_i)$
2 Confidence Loss $\mathcal{L}_{confidence}$ = $-\sum_{i=1}^{n} c_i - \left[\frac{1}{K}\sum_{k=1}^{K} I(M_k(r_i|q) == \overline{r_i})\right]$
3 Diversity Loss $\mathcal{L}_{diversity} = -H(\alpha_1, \ldots, \alpha_n)$
4 Language Model Loss $\mathcal{L}_{LM}$ = $-log\ p(y_{ans}|q, p_1, \ldots p_n)$
p_1 … p_n q
Blocked Bidirectional Attention Mask
α_1 … α_n
Passage Relevance
Relevance-Guided Attention Mask

Figure 4: The architectures of LKG-RALM.

To analyze the relationship between these patterns and RALM performance, we examined the correlation between attention weight sums and model accuracy. Figure 2(d) reveals that increased attention on relevant passages in edge-focused and uniform patterns yielded no performance gains, while middle-focused patterns demonstrated a strong positive correlation with RALM accuracy. Manipulation experiments further supported these findings: artificially replacing edge-focused and uniform patterns with middle-focused attention on relevant passages disrupted the model's attention structure, leading to performance degradation. As shown in Figure 3(c), deliberately redirecting middle-focused patterns to irrelevant passages significantly decreased performance, while concentrating this attention on relevant passages improved it. These results suggest that guiding middle-focused attention towards relevant passages could significantly enhance RALM's effectiveness.

4 Methodology

4.1 Overview

The vanilla attention of LLMs often suffers from distractibility and positional bias, which is unsuitable for open-QA with retrieved passages. We take advantage of layer-wise knowledge of LLMs to assess passage relevance, then guide the LLM's attention to generate answers, effectively mitigating these issues. The overall framework is illustrated in Figure 4.

4.2 Adding Special Tokens

To clearly delineate the boundary of the given question and each passage, we introduce trainable special tokens into the sequence. Specifically, we add $[d]$ and $[e_i]$ tokens as boundary markers at the beginning and end of each retrieved passage, respectively, while $[q]$ and $[e_q]$ tokens demarcate the question.

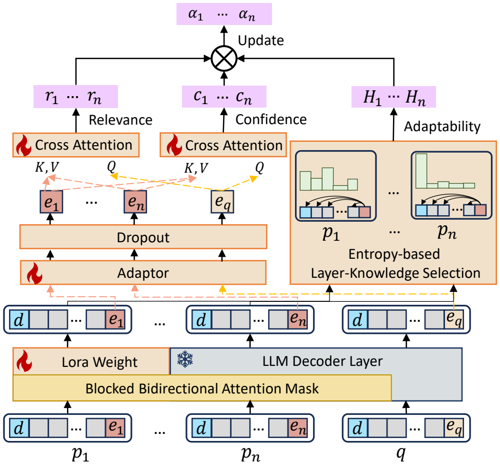


Figure 5: The layer-wise passage estimator.

4.3 Layer-wise Passage Estimator

Traditional passage estimators often rely on BERT-like structures, but these methods typically yield low accuracy and fail to leverage the rich, layer-specific knowledge embedded in LLMs. We propose a layer-wise passage estimator as Figure 5, which utilizes per-layer knowledge of LLM to assess relevance scores from multifaceted views, along with their associated confidence. Additionally, it incorporates an entropy-based layer-knowledge selection, which analyzes the attention distribution to determine the applicability of each layer's knowledge to the passages. By combining these comprehensive estimations, our approach provides trustworthy guidance for the LLM's attention.

4.3.1 Layer-wise Relevance and Confidence Estimation.

For a given layer l, we leverage the LLM's internal representations to compute relevance and confidence scores. To adapt the LLM's parameter to relevance assessment, we add trainable low-rank weights (LoRA) to each decoder layer. To enhance contextual understanding within passages, we follow previous work (BehnamGhader et al., 2024) to adopt Blocked Bidirectional Attention Mask rather than the causal attention. To aggregate sentence-level information, we extract the hidden states of the last special tokens as sentence embedding: e_i for each passage and e_q for question. An adapter and dropout components are then used to enhance their robustness. Finally, two cross-attention components compute the relevance scores $r_1, \dots, r_n$ and confidence scores $c_1, \dots, c_n$ between passages and the question, respectively.

4.3.2 Optimizing Estimator.

To optimize our layer-wise passage estimator, we introduce three specialized loss functions, each addressing a crucial aspect of effective relevance estimation:

Relevance Loss. To ensure the model accurately identifies relevant passages, we employ a relevance loss. This loss function encourages the estimated relevance scores to closely align with the ground truth, thereby improving the model's ability to distinguish between relevant and irrelevant passages:

$$L_{relevance} = -\frac{1}{n}\sum_{i=1}^{n}[\bar{r_i}\log(r_i)+(1-\bar{r_i})\log(1-r_i)] \tag{1}$$

where $\bar{r_i}$ is the ground truth label, and r_i is the estimated relevance score.

Confidence Loss Recognizing that not all relevance predictions are equally reliable, we introduce a confidence loss. We posit that the model should exhibit high confidence for easier samples to classify, while maintaining lower confidence for more challenging and confusing cases. To this end, we leverage external models (such as BGE (Chen et al., 2024)) to assist in determining sample difficulty:

$$L_{confidence}{=}{-}\sum_{i=1}^{n}(c_i{-}[\frac{1}{K}\sum_{k=1}^{K} I(M_k(r_i|q){==}\bar{r_i})]) \tag{2}$$

where c_i is the estimated confidence score, M_k represents K different external models, and $I(\cdot)$ is the indicator function. This loss trains the model to produce confidence scores that accurately reflect the trustworthiness of its relevance predictions:

Diversity Loss. To ensure a comprehensive utilization of layer-specific knowledge and avoid overly homogeneous relevance guidance, we employ a diversity loss based on the entropy of the final relevance guidance:

$$L_{diversity} = -H(\alpha_1, \dots, \alpha_n) \tag{3}$$

where $H(\cdot)$ is the entropy function, and $\alpha_1, \dots, \alpha_n$ are the final relevance guidance weights.

Combining these loss functions through simple addition, our estimator learns to provide accurate, confident, and diverse relevance assessments across different layers of the LLM. The Relevance Loss helps to quickly narrow down the search space to the most pertinent passages, while the Diversity Loss encourages a broader exploration of potentially relevant information, increasing the chances of recalling the correct answer. Although these two losses may seem antagonistic, their balanced combination leads to a more robust and comprehensive relevance assessment.

4.3.3 Entropy-based Layer-Knowledge Selection.

Inspired by Hyeon-Woo et al. (2023), to ensure the effective utilization of layer-specific knowledge in passage assessment, we propose an entropy-based layer-knowledge selection to identify which layers provide the most informative and contextually rich representations for each passage.

Specifically, for each passage p_i, we calculate the entropy H_i of the attention distribution from each passage's last special token to other tokens in the sequence:

$$H_i = -K\sum_{j=1}^{n} w_i^j \log w_i^j \tag{4}$$

where w_i^j denotes the attention weight from the last special token e_i to the j-th token in the sequence, n is token number of passage p_i, and K is the scaling factor. A higher entropy value indicates that the sentence embedding gathers a broader range of contextual information.

Finally, we use a selection weight to aggregate the layer-wise relevance and confidence scores to update the relevance guidance:

$$\alpha_i^l = \beta \left(\log(1 + H_i) \cdot \log(1 + r_i) \cdot \log(1 + c_i)\right) + (1 - \beta)\alpha_i^{l-1} \tag{5}$$

where α_i^l is the updated relevance guidance for passage p_i at layer l, and β balances current and previous layer assessments. To mitigate numerical oversensitivity, we employ a logarithmic multiplication. This approach combines layer-wise relevance and confidence estimation with entropy-based layer-knowledge selection, enabling our estimator to leverage diverse knowledge across LLM layers and provide robust guidance for the LLM's attention mechanism.

4.4 Relevance-aware Passage Fusion

To mitigate the issues of distractibility and positional bias, we propose a Relevance-aware Passage Fusion that selectively directs LLM attention to relevant passages based on the relevance guidance obtained from the Layer-wise Passage Estimator.

To effectively guide the LLM's attention towards relevant passages while mitigating the effects of distractibility and positional bias inherent in traditional attention frameworks, we introduce a relevance-guided attention mask. This mask dynamically modulates query-passage interactions based on estimated relevance, preserves intra-passage context, and inhibits cross-passage interference, thereby enhancing the model's capacity to prioritize salient information. The mask modulates the attention weights based on the estimated relevance of each passage. Formally, for each layer l, we define the attention mask M^l as:

$$M_{ij}^l = \begin{cases} \alpha_k^l, & \text{if } i \in q \text{ and } j \in p_k \\ & \text{(middle-focused attention heads)} \\ 1, & \text{if } i \in q \text{ and } j \in p_k \\ & \text{(other attention heads)} \\ 1, & \text{if } i \in p_k \text{ and } j \in p_k \\ & \text{(same passage)} \\ 0, & \text{if } i \in p_k \text{ and } j \in p_m \text{ where } k \neq m \\ & \text{(different passages)} \end{cases} \tag{6}$$

where q represents the set of query token positions, p_k is the set of token positions for passage k, and α_k^l is the relevance guidance for passage k at layer l. As our analysis of RALM attention, we selectively apply relevance-guided attention mask to Middle-focused attention heads only, while maintaining the functionality of Edge-focused and Uniform attention patterns. Finally, we use the standard language modeling loss to jointly fine-tune the LLM.

5 Experiments

5.1 Experimental Setting

5.1.1 Datasets

To assess performance across diverse data characteristics, we employ a range of representative datasets for RALM evaluation. These include Natural Question (NQ) (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), StrategyQA (Geva et al., 2021), HotpotQA (Yang et al., 2018), PopQA (Mallen et al., 2022), and 2WikiMQA (Ho et al., 2020). Detailed descriptions are provided in Appendix B.

5.1.2 Baselines

We categorize our baselines into three groups: closed-book LLM without retrieval, LLM with retrieval, and robust RALM. The first two groups include LLAMA-3.1 (Dubey et al., 2024), Qwen-2.5 (Yang et al., 2024), ChatGPT, GPT4, and Claude-3-Sonnet. The third group comprises REPLUGE (Shi et al., 2023b), Self-RAG (Asai et al., 2023), RA-ISF (Liu et al., 2024b) Noise-Resistant (Yoran et al., 2024), ChatQA-1.5 (Liu et al., 2024c) and RankRAG (Yu et al., 2024). Comprehensive descriptions of these baselines are provided in Appendix C.

5.1.3 Evaluation criteria

In evaluating the quality of the predicted answers, we employ the standard exact match (EM) metric on 5-shot, following previous work (Karpukhin et al., 2020; Izacard et al., 2022). The generated answer is first normalized by lowercasing, removing articles, punctuation, and duplicated whitespace. The EM score is binary for one question, with a value of 1 if the predicted answer matches the ground-truth answer exactly, and 0 otherwise. Then we averaged the EM scores across all questions in the test set and then multiplied by 100 to obtain final scores.

Table 1: Overall Performance. **Bold** numbers indicate the best score across all models, while underlined numbers represent the best score within each category.

Model	NQ	TriviaQA	HotpotQA	PopQA	2WikiMQA
General LLMs					
LLAMA-3.1-8B	18.7	78.5	16.5	22.1	13.9
+ Retrieval	30.9	70.7	26.0	34.9	9.6
+ Fine-tuning	35.7	77.4	28.9	37.1	25.3
LLAMA-3.1-70B	21.8	89.7	24.1	27.5	21.6
+ Retrieval	42.7	82.4	35.5	45.3	13.5
+ Fine-tuning	44.9	89.1	38.5	50.3	28.4
Qwen-2.5-7B	37.5	80.2	20.3	24.8	16.2
+ Retrieval	44.3	83.5	29.7	37.8	17.8
+ Fine-tuning	46.1	86.3	30.8	40.6	28.8
Qwen-2.5-72B	39.9	90.5	26.3	29.8	24.1
+ Retrieval	45.1	90.6	37.2	47.9	27.7
+ Fine-tuning	47.6	90.5	39.4	52.2	36.0
ChatGPT	38.6	82.9	29.9	28.4	23.9
+ Retrieval	46.7	79.7	31.2	49.9	27.2
GPT-4	40.3	87.0	34.5	31.3	29.8
+ Retrieval	40.4	75.0	27.6	44.3	14.4
Claude-3-Sonnet	49.2	87.5	32.8	33.4	31.4
+ Retrieval	55.1	**90.8**	33.3	52.4	32.6
Robust RALM					
REPLUGE	23.8	58.6	21.8	40.1	25.7
Self-RAG	28.4	61.6	25.4	44.8	30.2
RA-ISF	31.3	63.2	28.9	46.8	31.7
Noise-Resistant RALM	45.7	80.3	34.4	48.1	34.7
ChatQA-1.5	47.0	85.6	35.5	45.3	13.5
RankRAG	54.2	86.5	42.7	59.9	38.2
LKG-RALM					
LLAMA-3-8B	53.6	87.9	42.4	56.5	38.7
LLAMA-3-70B	59.9	89.5	44.7	62.0	41.1
LLAMA-3.1-8B	55.3	88.6	43.1	57.2	39.0
LLAMA-3.1-70B	61.0	89.9	45.8	62.6	41.3
Qwen-2.5-7B	55.4	88.1	43.1	57.4	39.7
Qwen-2.5-72B	**61.5**	90.0	**46.1**	**62.7**	**41.4**

5.2 Overall Performance

The results in Table 1 demonstrate varying performance across different model types. Closed-book LLMs show strong baseline performance but face limitations in their knowledge base. Adding retrieval generally improves performance, as seen with LLAMA-3.1-70B improving from 21.8 to 42.7 on NQ, and further fine-tuning brings additional gains (reaching 44.9). However, this improvement isn't consistent across all models and datasets. For instance, GPT-4 with retrieval shows decreased performance on TriviaQA (87.0 to 75.0).

The combination of retrieval and fine-tuning shows promising results, particularly for larger models. Qwen-2.5-72B benefits significantly from both enhancements, with performance on NQ improving from 39.9 (base) to 45.1 (+ retrieval) to 47.6 (+ retrieval & fine-tuning). Claude-3-Sonnet with retrieval achieves strong results, reaching 55.1 on NQ and 90.8 on TriviaQA. Robust RALM methods, particularly RankRAG, demonstrate effective utilization of retrieved passages, showing consistent improvements across datasets. RankRAG achieves strong performance with 54.2 on NQ and 59.9 on PopQA, outperforming many traditional retrieval-augmented approaches.

Table 2: Ablation result of LKG-RALM, where "LPR", "ELS", and "RPF" stand for Layer-wised Passage Relevance, Entropy-based Layer-Knowledge Selection, and Relevance-aware Passage Fusion, respectively.

Model Varient	NQ	TriviaQA	HotpotQA	PopQA	2WikiMQA
LKG-RALM-8B	55.3	88.6	43.1	57.2	39.0
w/o LPR	50.6	86.0	39.5	52.5	34.8
w/o ELS	53.1	87.7	41.8	55.8	37.2
w/o RPF	30.9	70.7	26.0	34.9	9.6
w/o Auxiliary Loss	54.2	87.3	42.3	56.4	37.8

LKG-RALM outperforms baseline models across all datasets. Compared to closed-book LLMs, it shows substantial gains of 12.3 percentage points on NQ to 10.0 on 2WikiMQA. When compared to retrieval-augmented and fine-tuned models, LKG-RALM still demonstrates superior performance, with Qwen-2.5-72B based LKG-RALM achieving the best results across most metrics (61.5 on NQ, 46.1 on HotpotQA). The performance gap between 8B and 70B variants (2.3-5.7 percentage points) suggests that larger models can better leverage our approach, particularly on complex tasks like PopQA.

5.3 Ablation Results

5.3.1 Effect of Designed Components

Table 2 shows the ablation results of LKG-RALM with LLAMA-3.1-8B as the backbone. All proposed components contribute significantly to the final performance. Replacing the Layer-wise Passage Estimator with a reranking model causes a substantial performance drop across all tasks, with an average decrease of 3.96. This highlights its crucial role in using layer-wised LLM knowledge to assess passage relevance. The Entropy-based Layer-Knowledge Selection mechanism proves effective, as its removal leads to an average EM decrease of 1.52, showing the importance of dynamically selecting informative layer representations for each passage. Ablating the Relevance-aware Passage Fusion component results in significant performance degradation, with an average EM decrease of 22.22. This demonstrates our approach's effectiveness in reducing distractibility and positional bias when processing multiple passages, compared to traditional attention mechanisms. Finally, the auxiliary losses improve performance across most tasks by 1.06, indicating their value in guiding the model to consider passage relevance, prediction confidence, and diverse utilization of layer knowledge during training.

5.4 Robustness Analysis

5.4.1 Increased Number of Retrieved Passages

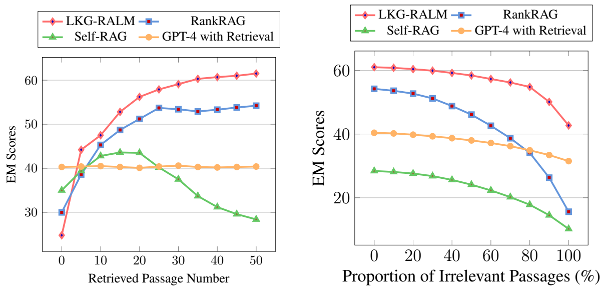
To assess the scalability and efficiency of LKG-RALM in handling larger amounts of retrieved information, we conducted experiments varying the number of retrieved passages from 0 to 50. Figure 6(a) illustrates the performance trends of LKG-RALM compared to baseline models across different datasets.

The experiment reveals that LKG-RALM demonstrates superior scalability and maintains high performance even as the number of retrieved passages increases significantly. LKG-RALM shows a steady increase in EM scores from 24.8 to 61.5 as the number of passages grows, with only a slight plateau effect beyond 35 passages, indicating effective utilization of additional information without suffering from information overload. In contrast, baseline models like RankRAG and Self-RAG initially show improvements with more passages, but their performance begins to degrade or plateau beyond 25 passages. RankRAG reaches a peak of 53.7 at 25 passages before slightly declining to 54.2 at 50 passages, while Self-RAG peaks at 43.5 at 25 passages before sharply declining to 28.4 at 50 passages. GPT-4 with Retrieval shows remarkable stability, maintaining a nearly constant performance (around 40.4) regardless of the number of passages, indicating its strong innate knowledge but potential limitations in effectively utilizing additional retrieved information.

LKG-RALM maintains its edge, with a 6.8 EM score advantage over RankRAG at 50 passages (61.0 vs 54.2). Despite the increased potential for irrelevant information with more passages, LKG-RALM's performance remains robust, underscoring the effectiveness of its relevance-aware passage fusion mechanism.

5.4.2 Higher Proportions of Irrelevant Passages

Our method internally attends attention to relevant passages for the given question, facilitating evidence-seeking from noisy contexts. To evaluate its robustness and noise tolerance, we conducted adversarial testing by incrementally replacing the 50 retrieved passages with irrelevant passages, ranging from 0% to 100% substitution.



(a) Impact of Increasing Passage Number

(b) Impact of Irrelevant Passages

Figure 6: Robustness to the number of retrieved passages and the proportion of irrelevant passages.

From Figure 6(b), LKG-RALM showed strong resilience against irrelevant information. When increasing irrelevant passages to 100%, the EM score only gradually decreased from 61.0 to 42.7, significantly outperforming other retrieval-based models. Even with 80% irrelevant input, LKG-RALM maintained a strong EM score of 54.8. In comparison, models without explicit relevance modeling like RankRAG saw sharp performance drops, falling from 54.2 to 15.6 with fully irrelevant passages. While GPT-4 with Retrieval showed high noise tolerance, dropping only from 40.4 to 31.5 under fully irrelevant conditions, it did not leverage relevant information as effectively as LKG-RALM, as shown in our earlier experiment. LKG-RALM's superior performance stems from its explicit relevance modeling, which helps it focus on pertinent information while filtering out noise. This allows it to effectively balance the use of retrieved knowledge with its inherent model capabilities.

6 Conclusion

In this work, we proposed LKG-RALM, which leverages layer-wise knowledge within LLMs to guide attention toward relevant passages, addressing distractibility and positional bias in handling retrieved passages. A layer-wise passage estimator evaluates passage relevance by utilizing diverse layer knowledge within the LLM. Entropy-based layer-knowledge selection dynamically identifies the most relevant layers for accurate passage assessment. Relevance-aware passage fusion selectively prioritizes crucial content, reducing the impact of irrelevant passages and overcoming positional bias. Extensive experiments across multiple datasets demonstrate that LKG-RALM achieves notable improvements in accuracy and robustness for knowledge-intensive tasks.

Limitations

Our work has several important limitations that should be acknowledged:

First, while our layer-wise passage estimator significantly improves RALM performance, it introduces additional computational overhead. The need to process passages through multiple layers for relevance assessment increases both memory usage and inference time. Although this overhead is relatively small compared to the base LLM inference, it may impact real-time applications or resource-constrained environments. Future work could explore more efficient methods for leveraging layer-wise knowledge without significant computational costs.

Second, our approach relies heavily on the quality of retrieved passages. While LKG-RALM shows improved robustness to irrelevant passages, its performance still degrades when the retrieval quality is poor or when dealing with queries requiring information beyond the knowledge cutoff date of the retrieval corpus. This limitation is particularly evident in rapidly evolving domains where the retrieved information may become outdated quickly.

Third, the effectiveness of our layer-knowledge selection mechanism may vary across different LLM architectures and sizes. While we demonstrated strong performance with LLAMA-3.1 and Qwen-2.5, the optimal configuration of layer-wise knowledge utilization might need to be adjusted for different model architectures. Additionally, our current approach to entropy-based layer selection may not capture all aspects of layer-specific knowledge representation.

Ethics Statement

Our work utilizes publicly available datasets and pre-trained language models, adhering to established data usage guidelines. However, several ethical considerations deserve attention. While LKG-RALM shows improved robustness in handling retrieved information, it inherits potential biases present in both the pre-trained language models and the retrieval corpus, which could affect the model's responses across different demographic groups or topic areas. We emphasize that our work primarily focuses on technical improvements in retrieval-augmented language modeling and should be complemented with dedicated bias mitigation strategies. Additionally, the improved performance of our model in handling retrieved passages raises questions about information authenticity and attribution. While LKG-RALM can better identify and utilize relevant information, users should be aware that the model's responses are based on retrieved passages that may contain inaccuracies or outdated information. We recommend implementing clear attribution mechanisms and confidence indicators in practical applications.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 62102277), the National Key R&D Program of China under Grant No. 2022YFC3303600, the Zhejiang Provincial Natural Science Foundation of China under Grant No. LY23F020010.

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*.

Manoj Ghuhan Arivazhagan, Lan Liu, Peng Qi, Xinchi Chen, William Yang Wang, and Zhiheng Huang. 2023. Hybrid hierarchical retrieval for open-domain question answering. In *Findings of ACL*, pages 10680–10689.

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*.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv*.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv*.

Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual error correction for abstractive summarization models. *arXiv*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul

Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv*.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv*.

Yao Fu. 2024. How do language models put attention weights over long context? *Yao Fu's Notion*.

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

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. " O'Reilly Media, Inc.".

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv*.

Hangfeng He, Hongming Zhang, and Dan Roth. 2022. Rethinking with retrieval: Faithful large language model inference. *arXiv*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv*.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv*.

Yangsibo Huang, Daogao Liu, Zexuan Zhong, Weijia Shi, and Yin Tat Lee. 2023. k nn-adapter: Efficient domain adaptation for black-box language models. *arXiv*.

Nam Hyeon-Woo, Kim Yu-Ji, Byeongho Heo, Dongyoon Han, Seong Joon Oh, and Tae-Hyun Oh. 2023. Scratching visual transformer's back with uniform attention. In *ICCV*, pages 5807–5818.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv*.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv*.

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.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv*.

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. *TACL*, 7:453–466.

Xianzhi Li, Xiaodan Zhu, Zhiqiang Ma, Xiaomo Liu, and Sameena Shah. 2023. Are chatgpt and gpt-4 general-purpose solvers for financial text analytics? an examination on several typical tasks. *arXiv*.

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

Shuai Liu, Hyundong J Cho, Marjorie Freedman, Xuezhe Ma, and Jonathan May. 2023. Recap: Retrieval-enhanced context-aware prefix encoder for personalized dialogue response generation. *arXiv*.

Yanming Liu, Xinyue Peng, Xuhong Zhang, Weihao Liu, Jianwei Yin, Jiannan Cao, and Tianyu Du. 2024b. Ra-isf: Learning to answer and understand from retrieval augmentation via iterative self-feedback. *arXiv*.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Mohammad Shoeybi, and Bryan Catanzaro. 2024c. Chatqa: Building gpt-4 level conversational qa models. *arXiv*.

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*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *NeurIPS*, 35.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv*.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. *arXiv*.

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

Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. In chatgpt we trust? measuring and characterizing the reliability of chatgpt. *arXiv*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023a. Large language models can be easily distracted by irrelevant context. In *ICML*, pages 31210–31227. PMLR.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023b. Replug: Retrieval-augmented black-box language models. *arXiv*.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. 2024. Repetition improves language model embeddings. *arXiv*.

Boxin Wang, Wei Ping, Peng Xu, Lawrence McAfee, Zihan Liu, Mohammad Shoeybi, Yi Dong, Oleksii Kuchaiev, Bo Li, Chaowei Xiao, et al. 2023a. Shall we pretrain autoregressive language models with retrieval? a comprehensive study. *arXiv*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023b. Improving text embeddings with large language models. *arXiv*.

Shangyu Wu, Ying Xiong, Yufei Cui, Xue Liu, Buzhou Tang, Tei-Wei Kuo, and Chun Jason Xue. 2024a. Improving natural language understanding with computation-efficient retrieval representation fusion. *arXiv*.

Zhenyu Wu, Chao Shen, and Meng Jiang. 2024b. Instructing large language models to identify and ignore irrelevant conditions. *arXiv*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv*.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *ICLR*.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *arXiv*.

Ningyu Zhang, Shumin Deng, Xu Cheng, Xi Chen, Yichi Zhang, Wei Zhang, Huajun Chen, and Hangzhou Innovation Center. 2021. Drop redundant, shrink irrelevant: Selective knowledge injection for language pretraining. In *IJCAI*, pages 4007–4014.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, et al. 2024. A comprehensive study of knowledge editing for large language models. *arXiv*.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. *arXiv*.

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*.

A Attention Analysis Setting

A.1 Model and Dataset Selection

Our analysis of attention patterns in Retrieval-Augmented Language Models (RALMs) was conducted using LLAMA-3.1-8B-instruct as the base model. We randomly selected a sample of 2000 queries from the Natural Questions and TraviaQA datasets, which consist of real-world queries submitted to Google Search and Allen Institute along with high-quality human-annotated answers extracted from Wikipedia pages.

A.2 Attention Pattern Classification

In our study, we focused on three main categories of attention patterns: edge-focused attention, uniform attention, and middle-focused attention. To analyze these patterns, we examined the attention distribution across all attention heads in the model. For each attention head, we calculated the percentage of attention allocated to different parts of the input sequence, specifically the beginning, middle, and end. Based on this distribution, we classified each attention head into one of the three main categories. We then quantified the prevalence of each attention pattern type across all attention heads to gain a comprehensive understanding of the model's attention behavior.

A.3 Correlation Analysis

To investigate the relationship between attention patterns and model performance, we conducted a correlation analysis. This involved calculating the sum of attention weights for each pattern type and measuring the model's accuracy on the test set. We then computed the correlation between these attention weight sums and the model's accuracy for each pattern type. This analysis allowed us to identify which attention patterns were most strongly associated with improved model performance.

A.4 Manipulation Experiments

To further validate our findings and explore the causal relationships between attention patterns and model performance, we conducted two types of manipulation experiments. In the first experiment, we artificially replaced edge-focused and uniform attention patterns with middle-focused attention on relevant passages. This allowed us to observe how redirecting attention to potentially more informative parts of the input affected the model's performance. In the second experiment, we deliberately redirected middle-focused attention patterns to both irrelevant and relevant passages. By comparing the model's performance under these different conditions, we were able to assess the impact of focused attention on specific parts of the input.

Throughout our analysis, we created detailed visualizations to illustrate the different attention patterns and their impact on model performance. These visualizations, presented in Figure 2 and Figure 3(b), provide a clear and intuitive representation of our findings, helping to elucidate the complex relationships between attention mechanisms and RALM performance. By systematically examining different types of attention patterns, their prevalence, and their relationship to model accuracy, we have identified potential areas for improvement in model design and training, particularly in guiding attention to relevant parts of the input for enhanced performance in open-domain question answering tasks.

B Dataset Description

To comprehensively evaluate the performance of Retrieval-Augmented Language Models (RALMs) across diverse data characteristics, we employ a range of representative datasets covering various aspects of question-answering tasks, from factoid questions to multi-hop reasoning and strategy-based inquiries. Below, we provide detailed descriptions of each dataset:

- *Natural Questions (NQ)*: Developed by Google Research (Kwiatkowski et al., 2019), this dataset comprises real-world queries submitted to Google Search, accompanied by high-quality human-annotated answers extracted from Wikipedia pages. NQ offers a rich mixture of long and short answer formats, reflecting authentic information-seeking behaviors across a broad range of subjects. Its structure, featuring both comprehensive passages and concise answer spans, provides a nuanced testing ground for RALMs.

- *TriviaQA*: Crafted by researchers at the Allen Institute for AI (Joshi et al., 2017), TriviaQA presents a formidable challenge with its extensive collection of question-answer pairs. These are sourced from trivia enthusiasts and paired with supporting evidence from Wikipedia and web searches. The dataset's hallmark is its high lexical and syntactic variance between questions and answers, necessitating robust retrieval and reasoning capabilities from models. By spanning both web and Wikipedia domains, it offers a comprehensive evaluation landscape.

- *StrategyQA*: Developed by the Allen Institute for AI (Geva et al., 2021), it focuses on multi-hop reasoning questions that demand implicit strategic thinking. StrategyQA's questions often require common sense reasoning, with answers typically being binary (yes/no) but necessitating complex cognitive processes. It is specifically designed to challenge and evaluate models' strategic thinking abilities, pushing the boundaries of AI reasoning.

- *HotpotQA*: A collaborative effort by Carnegie Mellon University (Yang et al., 2018), HotpotQA features Wikipedia-based question-answer pairs that explicitly require reasoning across multiple supporting documents. It includes sentence-level supporting facts for answer explanation and maintains a balance across different reasoning types, such as bridging and comparison. This structure makes HotpotQA particularly effective in assessing multi-hop reasoning capabilities.

- *PopQA*: Created by researchers at the University of Washington (Mallen et al., 2022), PopQA centers on questions about popular culture, including movies, music, celebrities, and current events. This dataset is crucial for testing models' ability to handle contemporary and rapidly evolving information. It challenges RALMs to navigate ambiguity and context-dependent information, reflecting the dynamic nature of real-world knowledge.

- *2WikiMQA*: Developed by the Graduate University for Advanced Studies (Ho et al., 2020), 2WikiMQA is a multi-hop open-domain question-answering dataset constructed from Wikipedia. It features questions that necessitate reasoning across multiple Wikipedia pages and includes complex queries that cannot be answered by a single fact. This dataset is designed to simultaneously test both retrieval accuracy and advanced reasoning capabilities of RALMs.

By employing this diverse set of benchmarks, we aim to provide a holistic assessment of model capabilities, from factual recall to complex reasoning and strategic thinking.

C Baseline Settings

Our baseline methods for open-QA tasks represent a diverse range of approaches, from pure language models to sophisticated retrieval-augmented systems. We categorize these baselines into three groups: closed-book LLM without retrieval, LLM with retrieval, and robust RALM. Each group showcases different strategies for tackling RAG challenges.

C.1 Closed-book LLM without retrieval and LLM with retrieval

The first two groups encompass state-of-the-art language models that have demonstrated exceptional capabilities in various natural language processing tasks. For models in the LLM with retrieval category, we employ a straightforward approach of concatenating retrieved content to the context, allowing the LLM to process the augmented input:

- *LLAMA-3.1* (2024) (Dubey et al., 2024): The latest iteration in the LLaMA series, LLAMA-3.1 builds upon 15 trillion texts, achieving the most effective open-source ability.

- *Qwen-2.5* (2024) (Yang et al., 2024): Developed by Alibaba, Qwen-2.5 represents a significant advancement in multilingual capabilities, trained on 18 trillion data to achieve state-of-the-art performance across various tasks.

- *ChatGPT* (2022): Developed by OpenAI, this model has gained widespread recognition for its conversational prowess and extensive knowledge base across diverse domains.

- *GPT-4* (2023) (Achiam et al., 2023): A large-scale, multimodal model developed by OpenAI, capable of accepting image and text inputs and producing text outputs. It exhibits human-level performance on various professional and academic benchmarks.

- *Claude-3-Sonnet* (2024): An advanced AI model from Anthropic, part of the Claude 3 model family, known for its strong performance across a wide range of tasks.

C.2 Robust RALM

The third group comprises advanced retrieval-augmented language models that enhance the robustness and effectiveness of RAG:

- *REPLUG* (2023) (Shi et al., 2023b): A retrieval-augmented language modeling framework that treats the language model as a black box and augments it with a tuneable retrieval model. It simply prepends retrieved documents to the input for the frozen black-box LM.

- *Self-RAG* (2023) (Asai et al., 2023): A framework that enhances an LM's quality and factuality through retrieval and self-reflection. It trains a single arbitrary LM that adaptively retrieves passages on-demand, and generates and reflects on retrieved passages and its own generations using special tokens.

- *RA-ISF* (2024) (Liu et al., 2024b): A framework that iteratively decomposes tasks and processes them in three submodules to enhance the model's problem-solving capabilities. It aims to improve factual reasoning capabilities and reduce hallucinations.

- *Noise-Resistant RALM* (2024) (Yoran et al., 2024): This approach focuses on making retrieval-augmented language models robust to irrelevant context. It proposes two methods: a simple baseline that filters out retrieved passages using an NLI model, and a method for automatically generating data to fine-tune the language model.

- *ChatQA-1.5* (2024) (Liu et al., 2024c): An evolution of the ChatQA model, this version introduces refinements aimed at enhancing effectiveness in question-answering tasks, particularly in conversational contexts.

- *RankRAG* (2024) (Yu et al., 2024): A instruction fine-tuning framework that instruction-tunes a single LLM for the dual purpose of context ranking and answer generation in RAG.

In our experimental setup, the RankRAG results are referenced from the original paper using LLaMA-3-70B in a zero-shot setting and are supported by the authors; ChatQA [1] leverages LLaMA-3-70B in a five-shot setting; Noise-Resistant RALM [2] is reproduced using LLaMA-3.1-8B; and RA-ISF [3] is implemented with ChatGPT-3.5.

D Implementation Details

Our model foundation utilizes LLAMA and Qwen. For retrieval, we follow ATLAS (Izacard et al., 2022) by using the Wikipedia dump from December 20, 2018, as our external corpus, comprising 28 million passages. We adopt a hybrid retrieval (Arivazhagan et al., 2023), where BM25 is grounded on the Elastic Search (Gormley and Tong, 2015), while the dense retriever is based on the FAISS index (Johnson et al., 2019). The training data is followed by Self-RAG (Asai et al., 2023). The trainable low-rank weights were implemented using LoRA (Hu et al., 2021), with a rank dimension of 256. The hidden size of the adaptor is set to 4096. We optimized all trainable parameters using the AdamW optimizer with a learning rate of 1e-5. The batch size was set to 32, and a warmup ratio of 0.1 was employed along with a cosine learning rate scheduler. Three external relevance scores are obtained from BGE-M3 [4], E5-mistral-7b-instruct [5], GTE-Qwen2-7B-instruct [6]. The updating factor β for layer-wised relevance guidance was set to 0.2.

Notice that we can use separate LLMs for passage estimation and answer generation in parallel. A lighter estimator (e.g., 1.5B) paired with a larger generator (e.g., 8B) minimizes overhead, where the generator can share relevance guidance across some layers due to differing layer counts.

For the attention pattern analysis, we define the first 3 tokens as the head and the last 3 tokens as the tail, with the remaining tokens classified as the middle. We employ threshold-based metrics to distinguish between attention patterns:

- Edge-focused: Combined attention weights of head and tail exceed 75%.

- Uniform: Middle attention weights exceed 90%, with over 40% of tokens having attention weights greater than 1/(input length), and no single token's attention weight exceeding 10%. The term "uniform" is somewhat hyperbolic. What it actually represents is a pattern where no single token receives exceptionally high attention.

- Middle-focused: Middle attention weights exceed 90%, with either one or two tokens having attention weights above 30%, or three or more tokens having attention weights above 10%.

E Performance on general NLP tasks

Beyond open-QA, we assess the capabilities of LKG-RALM architecture on broader NLP benchmarks. Specifically, we evaluate on Multitask Language Understanding (MMLU) and Language

[1] https://huggingface.co/nvidia/Llama3-ChatQA-1.5-70B

[2] https://huggingface.co/datasets/Ori/strategyqa-ret-robust

[3] https://github.com/OceannTwT/ra-isf

[4] https://huggingface.co/BAAI/bge-m3

[5] https://huggingface.co/intfloat/e5-mistral-7b-instruct

[6] https://huggingface.co/Alibaba-NLP/gte-Qwen2-7B-instruct

Model	Hum.	Social.	STEM	Other	All
LLAMA-3-8B	73.8	75.2	69.5	73.5	73.0
ChatGPT	71.2	73.6	65.8	69.4	70.0
GPT4	85.7	87.9	84.2	87.8	86.4
Self-RAG	64.5	65.8	63.1	65.4	64.7
RankRAG	74.1	75.6	70.8	73.5	73.5
LKG-RALM-8B	75.3	77.2	71.9	74.8	74.8

Table 3: Performance on MMLU task.

Modeling, standing challenging tasks covering both understanding and generation.

E.1 MMLU

We evaluated LKG-RALM on the Multi-task Language Understanding (MMLU) benchmark (Hendrycks et al., 2020), a comprehensive multiple choice QA dataset consisting of 57 natural language understanding tasks, including elementary mathematics, US history, computer science, law, and more. Following previous work (Shi et al., 2023b), We grouped these tasks into four categories: Humanities, Social Science, STEM, and Other. We still use the Wikipedia dump as an external corpus for retrieving information to improve the performance on the MMLU task.

As shown in Table 3, the results demonstrate that LKG-RALM outperforms the original LLAMA model by a significant margin across all tasks. Specifically, we observe an average accuracy improvement of 1.5% on Humanities, 7.2% on Social Science, 2.8% on STEM, and 4.4% on other tasks over LLAMA-3.1-8B. Moreover, compared to other models, we have achieved competitive performance. LKG-RALM-8B outperforms ChatGPT by 4.8% on average and surpasses Self-RAG by 10.1%. Compared with RankRAG, we obtain 1.3 absolute improvements on average. This substantial performance boost can be attributed to two key factors. Firstly, the retrieved passages from Wikipedia provide useful external knowledge and context for the model to better understand the input texts. Secondly, the relevance-guided architecture enables more effective encoding and reasoning over lengthy context-like passages, facilitating passage understanding for solving complex MMLU tasks.

E.2 Language Modeling

As a crucial touchstone for evaluating general language generation capabilities, we assess the LKG-RALM architecture on language modeling benchmarks spanning diverse domains including websites, academic writing, code, and dialogue on the Pile dataset. These benchmarks require predicting subsequent tokens based on preceding textual context, and evaluating model fluency, coherence and grounding. A key challenge arises from lengthy context segmentation across long documents, which hinders encoding the full history to produce logically consistent continuations. To enable relevance-aware passage fusion component, we segment lengthy sequences into 100-token passages.

Model	# Params	Original	+LKG-RALM	Gain %
GPT-2	117M	1.33	1.22	8.27
	345M	1.20	1.13	10.83
	774M	1.19	1.14	4.20
	1.5B	1.16	1.01	12.93
Qwen-2.5	7B	0.95	0.90	5.26
	14B	0.88	0.84	4.54
	72B	0.70	0.66	5.71
LLAMA-3.1	8B	0.97	0.93	4.12
	70B	0.72	0.70	2.77

Table 4: Performance on language modeling task

Following prior work (Shi et al., 2023b), we report the standard bits-per-byte (BPB) metric which measures cross-entropy reduction to evaluate perplexity improvements. A smaller value of BPB means better performance. As shown in Table 4, LKG-RALM substantially enhances base LLMs like GPT-2, Qwen, and LLAMA across all categories of the Pile benchmark by 9.06% (GPT-2), 5.17% (Qwen), and 3.45% (LLAMA) BPB on average. This demonstrates LKG-RALM successfully encodes rich multi-granularity semantics to produce logical and human-like text continuations. The significant perplexity reductions validate that hierarchical encoding mechanisms enhance the language model's context capacity to track lengthy precedings for coherent generation. By effectively navigating long-range dependencies, LKG-RALM generates higher-quality and better-grounded natural language.

F Efficiency and Accuracy Trade-off

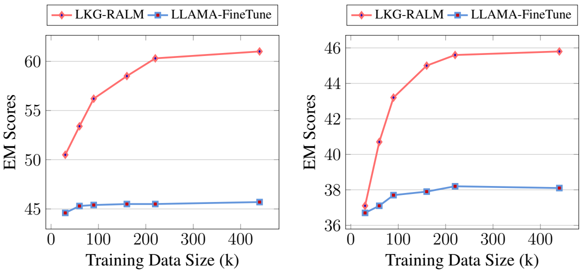
To contextualize our method's efficiency, we compared LKG-RALM's performance with existing models in Table 5. Self-RAG and RA-ISF require multiple rounds of retrieval and question decomposition-based multi-turn dialogue, respectively. Their low parallelism results in approximately 3.7x inference time compared to LLAMA-3.1-8B, with Self-RAG taking 3.07 seconds per query and RA-ISF requiring 3.44 seconds per query.

Table 5: Efficiency and Accuracy Trade-off for LKG-RALM and Baseline Models under 1024 Context Tokens.

Model	EM	Speed (s/query)	TFLOPs
Self-RAG	28.4	3.07	20.5
RA-ISF	31.3	3.44	63.8
Robust-RALM	45.7	0.74	14.6
RankRAG	54.2	1.25	145.4
LKG-RALM with Different Passage Estimator			
LLAMA-3.1-8B	30.9	0.82	16.3
+ Qwen-2.5-500M	53.6	0.83	18.1
+ Qwen-2.5-1.5B	54.8	0.85	20.0
+ Qwen-2.5-7B	55.3	0.91	30.9
LLAMA-3.1-70B	42.7	1.24	142.6
+ Qwen-2.5-500M	60.0	1.24	144.4
+ Qwen-2.5-1.5B	60.7	1.25	146.3
+ Qwen-2.5-7B	61.0	1.27	157.2

Noise-Resistant-8B and RankRAG-70B use BERT-based NLI models and Reranking Models, respectively, to assist in passage filtering or sorting. This introduces an additional 0.8% inference latency and 1.9% computational overhead for RankRAG-70B, increasing its processing time to 1.25 seconds per query and its computational cost to 145.4 TFLOPs for 1024 tokens. Similarly, LKG-RALM-70B employs the Qwen-2.5-7B model for passage relevance analysis, resulting in a 2.4% increase in inference latency (from 1.24 to 1.27 s/query) and a 10.2% increase in computational cost (from 142.6 to 157.2 TFLOPs). The low latency is attributed to the layer-wise passage estimator's ability to operate in high parallelism with LLM inference.

To further evaluate the trade-off between efficiency and accuracy, we conducted experiments using different LLM sizes for the layer-wise passage estimator. Table 5 shows that increasing the size of the passage estimator from Qwen-2.5-500M to Qwen-2.5-7B yields consistent improvements in EM scores for both LLAMA-3.1-8B and LLAMA-3.1-70B base models. For LLAMA-3.1-8B, the EM score improves from 53.6 to 55.3 as we scale up the estimator, with a modest increase in processing time from 0.83 to 0.91 seconds per query. The computational cost rises from 18.1 to 30.9 TFLOPs. Notably, even with the largest Qwen-2.5-7B estimator, LKG-RALM-70B maintains competitive efficiency compared to RankRAG-70B (1.27 vs 1.25 s/query) while achieving superior EM scores (61.0 vs 54.2). LKG-RALM's flexible framework allows users to balance accuracy and efficiency by selecting appropriate estimator sizes. For example, Qwen-2.5-1.5B with LLAMA-3.1-70B improves EM score by 0.7 over the 500M version, with minimal increases in query time and computational cost.



(a) NQ

(b) HotpotQA

Figure 7: Effects of training data size.

G Effects of Training Data Size

We conducted an analysis to understand how the scale of training data affects the model's performance. Specifically, we randomly sampled 30k, 60k, 90k, 160k, and 220k instances from our original 440k training instances and fine-tuned five LKG-RALM-70B variants on these subsets. We then compared the model performance on NQ and HotpotQA with our final LKG-RALM trained on the full 440k instances. We also evaluated LLAMA-3.1-70B fine-tuned on the same data subsets as a baseline. Figure 7 shows the models' performance trained on different amounts of data. For NQ, LKG-RALM-70B's accuracy improves from 50.5 with 30k training instances to 61.0 with the full 440k dataset. In contrast, LLAMA-3.1-FineTune shows minimal improvement, from 44.6 to 45.7 with a mere 1.1 increase. The performance gap between LKG-RALM-70B and LLAMA-3.1-FineTune widens significantly, from 5.9 at 30k instances to 15.3 at 440k instances.

The model's strong performance largely comes from its effective pre-training parameter space. With only 30k fine-tuning examples, LKG-RALM-70B shows impressive results, reaching 50.5 EM on NQ and 37.1 on HotpotQA. This indicates that minimal fine-tuning data can activate the model's core capabilities in passage estimation and robustness. The performance difference between LKG-RALM-70B and LLAMA-3.1-FineTune is clear even with limited data, showing our approach's effectiveness. As training data increases, LKG-RALM-70B's accuracy steadily improves, though gains slow after 220k examples. For NQ, the improvement from 220k to 440k is just 0.7, versus 1.8 from 160k to

Table 6: Distribution of Distinct Relevance Scores Across Model Layers

Number of Scores	1	2	3	4	5	6	7	Others
Percentage (%)	4.2	17.2	21.8	35.9	10.4	5.6	3.2	1.7

Table 7: Model Performance with Different Numbers of Relevance Scores

Model	NQ	TriviaQA	HotpotQA	PopQA	2WikiMQA
LLAMA-3.1-8B	18.7	78.5	16.5	22.1	13.9
+ Retrieval & Fine-tuning	35.7	77.4	28.9	37.1	25.3
+ LKG-RALM (k=1)	36.4	77.3	29.6	38.6	26.2
+ LKG-RALM (k=2)	40.7	78.5	31.3	42.3	26.5
+ LKG-RALM (k=3)	47.2	85.4	37.4	48.0	33.4
+ LKG-RALM (k=4)	50.1	86.0	39.3	51.6	34.6
+ LKG-RALM (k=5)	51.6	86.0	40.5	52.3	35.0
+ LKG-RALM (k=6)	51.9	86.1	40.5	53.2	35.4
+ LKG-RALM (k=7)	51.8	86.2	40.1	53.3	35.5
+ LKG-RALM (all)	55.3	88.6	43.1	57.2	39.0

220k. This shows that while more data helps performance, the benefits decrease with larger datasets. The model's architecture and pre-training are key to its success, enabling strong results with limited fine-tuning data, while additional training data has diminishing returns.

H Impact of Relevance Score Quantity

To investigate how the number of relevance scores affects model performance, we conducted experiments that preserved only the first k distinct relevance scores (k ranging from 1 to 7). Any subsequent relevance scores that were either similar or appeared later in the sequence were overwritten by these values. In our analysis of a 32-layer LLMAMA-3.1-8B, we observed significantly different relevance scores (L1 distance >0.3) distributed across layers. The distribution of these distinct scores is presented in Table 6.

The impact of varying k values on model performance across different datasets is shown in Table 7. The result reveals that LKG-RALM's accuracy consistently improved across metrics as more relevance scores were incorporated into the attention guidance mechanism. Notably significant performance improvements were observed at several key transitions. When increasing from k=2 to k=3, we observed substantial gains across all datasets, with NQ accuracy improving by 6.5 points (from 40.7 to 47.2) and TriviaQA showing a remarkable 6.9-point increase (from 78.5 to 85.4). The transition from k=3 to k=4 brought further improvements, particularly in NQ (2.9 points) and PopQA (3.6 points). Interestingly, while incremental improvements continued beyond k=4, they became more modest, with gains typically under 1.5 points per step.

The most striking performance boost was achieved when utilizing all relevance scores instead of limiting to the first seven distinct scores. This configuration led to substantial improvements across all datasets: NQ improved by 3.5 points (from 51.8 to 55.3), TriviaQA by 2.4 points (from 86.2 to 88.6), and 2WikiMQA by 3.5 points (from 35.5 to 39.0). These results strongly suggest that while the first few distinct relevance scores contribute significantly to model performance, the additional nuanced guidance from higher layers plays a crucial role in maximizing the model's capabilities.

While our experimental design focused on retaining the first k relevance scores, this approach may not be optimal as higher-layer knowledge representations could provide essential guidance for complex reasoning tasks. The superior performance achieved when utilizing all attention guidance signals validates their collective importance in enhancing model accuracy and demonstrates the value of maintaining a diverse set of relevance scores across different layers of the model architecture.