Rethinking the Necessity of Adaptive Retrieval-Augmented Generation through the Lens of Adaptive Listwise Ranking

Jun Feng[1], Jiahui Tang[2], Zhicheng He[2], Hang Lv[2], Hongchao Gu[2], Hao Wang[2], Xuezhi Yang[1*], Shuai Fang[1]
[1]Hefei University of Technology, [2]University of Science and Technology of China

***Abstract*—Adaptive Retrieval-Augmented Generation aims to mitigate the interference of extraneous noise by dynamically determining the necessity of retrieving supplementary passages. However, as Large Language Models evolve with increasing robustness to noise, the necessity of adaptive retrieval warrants re-evaluation. In this paper, we rethink this necessity and propose AdaRankLLM, a novel adaptive retrieval framework. To effectively verify the necessity of adaptive listwise reranking, we first develop an adaptive ranker employing a zero-shot prompt with a passage dropout mechanism, and compare its generation outcomes against static fixed-depth retrieval strategies. Furthermore, to endow smaller open-source LLMs with this precise listwise ranking and adaptive filtering capability, we introduce a two-stage progressive distillation paradigm enhanced by data sampling and augmentation techniques. Extensive experiments across three datasets and eight LLMs demonstrate that AdaRankLLM consistently achieves optimal performance in most scenarios with significantly reduced context overhead. Crucially, our analysis reveals a role shift in adaptive retrieval: it functions as a critical noise filter for weaker models to overcome their limitations, while serving as a cost-effective efficiency optimizer for stronger reasoning models. Our code is available at USTC-StarTeam/adaptive-listwise-ranking-rag.**

***Index Terms*—Retrieval-Augmented Generation, Adaptive Retrieval, Listwise Ranking, Knowledge-Intensive QA**

I. Introduction

Retrieval-Augmented Generation (RAG) stands as a pivotal paradigm that augments Large Language Models (LLMs) by incorporating external knowledge [1]–[4]. By grounding generation in retrieved passages, RAG effectively mitigates hallucinations [5] and facilitates continuous knowledge updates [6]. However, its efficacy is not guaranteed; recent studies [7], [8] show that indiscriminate retrieval often introduces extraneous noise. While standard RAG systems typically retrieve a fixed number of passages, this static strategy struggles to balance information coverage against noise intrusion: retrieving too few passages risks missing relevant information, while retrieving too many introduces noise that degrades model performance.

To address this dilemma, adaptive retrieval [9] has emerged as a dominant solution. By dynamically determining when and what to retrieve, it aims to shield models from irrelevant contexts and has historically yielded significant gains. However, the landscape of LLMs is rapidly evolving. Recent work [10] reveals that modern foundation models are becoming increasingly robust to retrieval noise, capable of ignoring irrelevant information via strong internal attention mechanisms. This evolution raises a fundamental question: *Is adaptive retrieval still essential for these increasingly capable RAG systems?*

To investigate this question, we propose AdaRankLLM, a framework that synergizes Adaptive Ranking with Instruction Distillation. Our approach hinges on a rigorous mechanism to determine retrieval necessity: an adaptive ranker equipped with a Passage Dropout mechanism. Utilizing a zero-shot listwise prompt, this module evaluates passage relevance, explicitly discarding irrelevant candidates or outputting a termination token when no passages are relevant. To ensure this verification capability is not confined to cost-prohibitive large models, we further integrate a two-stage progressive distillation paradigm. By leveraging clustering-based sampling and data augmentation, we transfer listwise ranking and adaptive filtering skills from teacher models to smaller open-source backbones, allowing us to verify the utility of adaptive retrieval across varying model capabilities and deployment constraints.

We conduct extensive experiments on three benchmarks across eight LLMs, spanning models from smaller open-source backbones to advanced reasoning-enhanced architectures. The results demonstrate that AdaRankLLM consistently achieves optimal performance in most scenarios while significantly reducing context overhead. Crucially, our analysis reveals a clear distinction: for weaker models, adaptive retrieval acts as a critical *noise filter* to ensure generation quality; for advanced models, while it offers limited performance gains, it serves as a vital *efficiency optimizer* by significantly reducing inference costs. Our main contributions can be summarized as follows:

- We examine the necessity of adaptive RAG for modern LLMs, moving beyond the traditional view to evaluate its distinct utility across models with varying capabilities.
- We propose AdaRankLLM, a framework that employs an adaptive ranker with a passage dropout mechanism and instruction distillation to endow smaller models with precise noise-filtering capabilities.
- Extensive experiments across eight LLMs reveal a functional distinction: adaptive retrieval acts as a critical noise filter for weaker models to ensure quality, while serving as a vital efficiency optimizer for advanced models.

*Corresponding author: xzyang@hfut.edu.cn

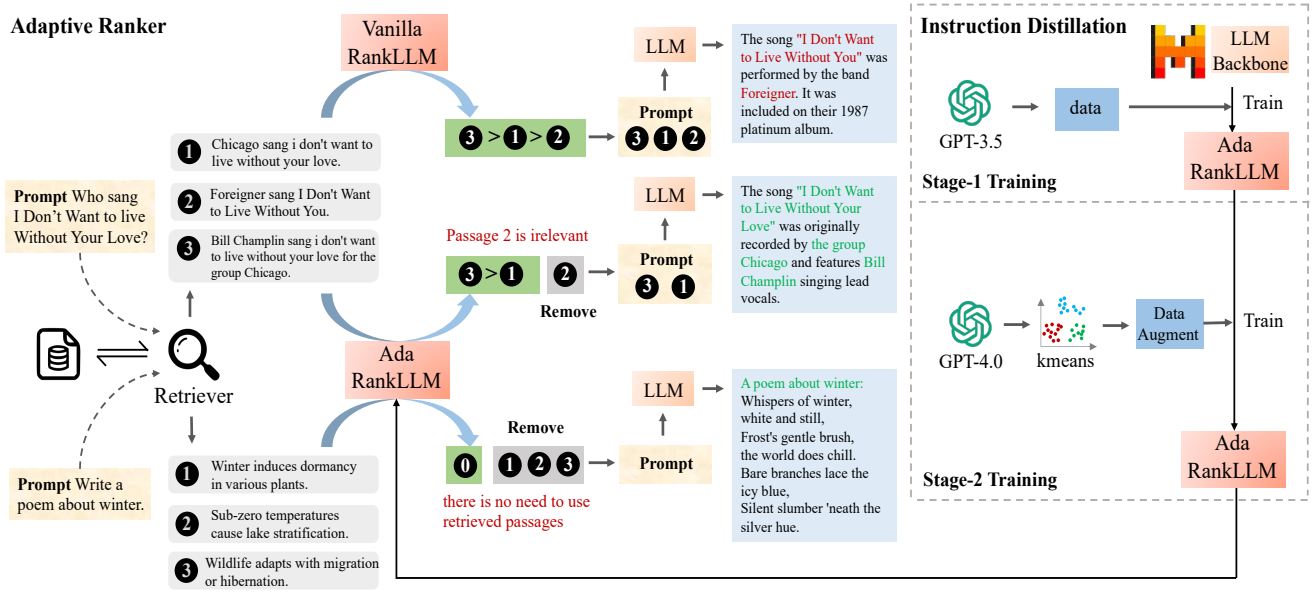

Fig. 1. The framework of AdaRankLLM. The left part shows two examples to demonstrate how the Adaptive Ranker works. The right part outlines the process of data collection and model training in order to distill an efficient ranker.

II. Related Work

Passage Ranking. Traditional ranking methods mainly rely on pointwise [11] or pairwise [12] scorers, yet they often overlook passage interrelations, prompting the shift to LLM-based listwise approaches [13]–[15]. While early distillation attempts like RankVicuna [16] and RankZephyr [17] successfully transferred listwise capabilities to 7B models, they remained constrained by the quadratic complexity of decoder-only architectures. Recent advancements address these limitations: LiT5 [18] reintroduces the Fusion-in-Decoder (FiD) architecture for linear scaling, while Rank-R1 [19] integrates Chain-of-Thought (CoT) reasoning via Group Relative Policy Optimization (GRPO) to explicitly justify relevance. Further reducing latency, non-generative approaches utilize passage embeddings [20] or attention patterns [21] directly. Our method aligns with this trajectory, enhancing zero-shot listwise distillation with adaptive capabilities.

Adaptive Retrieval. Recent studies [7], [8], [22], [23] indicate that RAG performance is easily compromised by extraneous information. Early approaches like FLARE [24] and Self-RAG [9] relied on token probabilities or learned reflection tokens, often resulting in high inference or fine-tuning costs. Recent research pivots toward internal state monitoring: CTRLA [25] and TARG [26] leverage "honesty" directions and prefix logits to gate retrieval without auxiliary classifiers. For complex reasoning, FAIR-RAG [27] , REPAIR [28] and RAPID [29] introduce structured, plan-aware loops to trigger "mid-course" retrieval based on intermediate steps rather than the initial query. Additionally, cost-aware reinforcement learning strategies [30] dynamically adjust retrieval depth. However, their complexity hampers zero-shot generalization and deployment efficiency, motivating our lightweight framework.

III. Method

A. Problem Definition

Given a user query x and a set of candidate passages $P = \{p_1, p_2, ..., p_m\}$ retrieved from a massive corpus $\mathcal{C}$ (where m is the number of candidates), the goal of Adaptive Retrieval-Augmented Generation is to identify an optimal ordered subset $A \subseteq P$. This process aims to maximize the relevance of the selected passages to x while filtering out noise. Crucially, the size of the result set $|A|$ is dynamic and dependent on the query complexity. The set A can range from the empty set $\emptyset$ (when retrieval is unnecessary) to the full candidate set. Finally, the selected passages A and query x serve as input to the generator to produce the response $y = \text{LLM}(A; x)$.

B. Adaptive Ranker with Passage Dropout

To transform the coarse candidate set P into the refined adaptive set A, we propose an **Adaptive Ranker** that advances the paradigm of listwise reranking. Unlike conventional rerankers that merely permute the fixed candidate list, our module integrates a passage dropout mechanism to dynamically filter extraneous noise while capturing the interrelations among passages.

Specifically, we design a zero-shot conversational prompt to instruct the LLM to perform relevance assessment and selection simultaneously. Each passage in the candidate set P is assigned a unique identifier (e.g., '[1]'). Instead of forcing a full permutation, the LLM is directed to select only the passages that contribute to answering the query and rank them by relevance. Crucially, the passage dropout mechanism enables a rejection option: if the model determines that a passage (or even the entire set) is irrelevant, it explicitly excludes it from the output list A. In extreme cases where no relevant

TABLE I
THE THREE DATASETS EMPLOYED IN THE EXPERIMENTS.

Datasets	Corpus(#passages)	Question Type	Retriever
ASQA	Wikipedia(21M)	Factoid(ambiguous)	GTR
QAMPARI	Wikipedia(21M)	Factoid(list)	GTR
ELI5	Sphere(899M)	Why/How/What	BM25

information is found, the model outputs a special termination token '[0]'. The specific prompt structure is illustrated in Figure 2. Following this adaptive ranking process, we parse the generated identifier list to construct the final context set $A = \{p_{r_1}, ..., p_{r_n}\}$, where the subset size n is dynamically determined by the model's relevance judgment.

C. Instruction Distillation

While proprietary LLMs like GPT-4 demonstrate exceptional zero-shot adaptive ranking capabilities, their prohibitive costs hinder direct large-scale deployment. Consequently, we aim to distill these capabilities into a compact open-source model using a progressive two-stage paradigm to balance distillation cost and data quality.

Stage 1: Structural Alignment. We utilize GPT-3.5 to generate a large-scale dataset from MSMARCO V1. This cost-effective initialization teaches the student model the fundamental listwise output schema and basic relevance ordering, establishing a solid structural foundation before.

Stage 2: Adaptive Refinement. To inject precise passage dropout capabilities, we rely on GPT-4's superior reasoning. To minimize costs, we apply K-means clustering to query embeddings to sample a representative subset. GPT-4 annotates these selected queries to fine-tune the student's decision boundaries for distinguishing valid evidence from noise.

Finally, by integrating the adaptive ranker with passage dropout and instruction distillation, we will get an AdaRankLLM that is capable of selecting necessary passages, while maintaining efficiency and accessible.

IV. EXPERIMENTAL SETUP

A. Datasets

We validate our method on three diverse public datasets: **ASQA** [31], **QAMPARI** [32], and **ELI5** [33]. Consistent with standard protocols [34], we employ Exact Match (EM) for ASQA, F1 score for QAMPARI, and Claim Recall for ELI5. To evaluate comprehensive capabilities, we also report an *Overall** metric computed as the average of these scores. Detailed dataset information are provided in Table I.

B. Baselines

We compare AdaRankLLM with several baselines categorized into three groups: (1) **Vanilla-k**: Models utilizing retrieved passages, where k denotes the number of passages provided to the LLM. (2) **Rerank-k**: Models employing reranked passages, with k indicating the number of reranked passages supplied to the LLM. RankGPT4 [35] is used as the reranker.

Prompt used for AdaRankLLM

```
<|im_start|> system
You are RankLLM, an intelligent assistant that can choose
passages based on their relevancy to the query.
<|im_end|>

<|im_start|> user
I will provide you with {num} passages, each indicated by
a numerical identifier[].Choose the passages based on
their relevance to the query: {query}.
[1] {passage 1}
[2] {passage 2}
...
[{num}] {passage {num}}
Search Query: {query}
Choose among the {num} passages above based on their
helpfulness to answer the search query. The passages
should be listed in descending order using identifiers.
The most relevant passages should be listed first. The
output format should be [] > [], e.g., [4] > [2]. If none
of the passages is helpful to answer the query, please
only response the '[0]'. Only respond with the ranking
results, do not say any word or explain. <|im_end|>

<|im_start|> assistant
[4] > [1] > [2]
<|im_end|>
```

Fig. 2. Illustration of the prompt template used in AdaRankLLM for relevance-based passage selection and reranking.

(3) **AdaRank-GPT4**: Models utilizing our proposed adaptive ranking prompt (Section III-B) with GPT-4, serving as a baseline to verify the necessity of adaptive RAG. Additionally, we report the **Oracle** values for each backbone, representing the theoretical optimal performance when selecting the best number of reranked passages ($k \in [0, 10]$).

C. Models

To comprehensively investigate the necessity of adaptive retrieval across varying model capabilities, we conduct experiments on seven backbone models: Alpaca-7b, Mistral-7b, GPT-3.5, GPT-4o, Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct, and Qwen3-8B (specifically evaluated with `enable_thinking=False` and `True`). These models span a diverse spectrum of open-source and proprietary architectures.

D. Implementation Details

a) Training Details: We fine-tune AdaRankLLM on Mistral-7B-v0.1 [36] using a subset of the MSMARCO V1 [37] dataset. The data comprises 100K queries with the top-20 BM25-retrieved passages, truncated to 10 to align with typical RAG settings. We use all 100K queries for the first stage and sample 5K queries for the second stage using **ADA2** embeddings [38] with k-means clustering ($k = 20$). Training runs for three epochs per stage with a learning rate of 5e-6 (cosine decay) and a batch size of 64.

b) Evaluation Details: For all three datasets, We utilize the pre-processed first-stage retrieval results from ALCE [34][1] and apply official chat templates for all backbone models. Regarding metric implementation, we strictly follow the ALCE codebase: employing STR-EM for ASQA and the T5-11B based TRUE model [39] to verify claim entailment for ELI5.

[1]https://huggingface.co/datasets/princeton-nlp/ALCE-data

TABLE II

RESULTS FOR ALPACA-7B, MISTRAL-INSTRUCT, GPT-3.5, GPT-4O, LLAMA, QWEN2.5-7B, AND QWEN3-8B. FOR THINKING MODEL QWEN3-8B, WE EXPLICITLY EVALUATE TWO CONFIGURATIONS: `ENABLE_THINKING=FALSE` AND `ENABLE_THINKING=TRUE`. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLD**, AND THE SECOND-BEST RESULTS ARE UNDERLINED.

Method	ASQA	ELI5	QAMPARI				ASQA	ELI5	QAMPARI			
	EM	Claim	Prec.	Rec.	F1	Overall*	EM	Claim	Prec.	Rec.	F1	Overall*
	Alpaca-7B						**Mistral-Instruct**					
Vanilla-0	19.03	10.17	21.53	5.81	7.96	12.55	27.61	15.80	12.87	13.79	11.58	18.33
Vanilla-1	31.09	12.33	26.21	8.72	11.07	18.16	36.30	13.83	22.01	10.08	11.58	20.57
Vanilla-3	32.70	13.73	26.12	9.82	11.85	19.43	44.46	14.83	21.24	11.87	12.69	24.05
Vanilla-5	32.75	13.33	25.21	10.03	11.93	19.34	46.23	14.70	21.66	14.52	14.27	25.07
Vanilla-10	29.99	12.70	25.62	9.04	11.34	18.01	47.23	15.43	21.14	16.70	15.40	26.02
Rerank-1	35.01	13.10	**31.95**	10.40	13.19	20.43	43.17	13.83	**28.26**	12.09	13.71	23.57
Rerank-3	35.03	**14.13**	29.00	11.28	13.57	**20.91**	47.36	14.83	25.84	14.66	15.52	25.90
Rerank-5	33.85	13.43	27.76	10.95	12.97	20.08	47.55	14.70	23.24	16.20	15.96	26.07
Rerank-10	29.51	12.57	24.99	9.20	11.41	17.83	46.81	15.43	20.19	16.78	15.21	25.82
AdaRankGPT4	**35.22**	12.93	31.29	**11.51**	**14.18**	20.78	46.66	**16.03**	25.39	**17.48**	**17.22**	26.64
AdaRankLLM	33.86	13.07	28.70	11.06	13.53	20.15	**48.14**	15.23	24.05	17.18	16.77	**26.71**
Oracle	46.78	25.77	-	-	27.49	33.35	64.53	28.93	-	-	33.23	42.23
	GPT-3.5						**GPT-4o**					
Vanilla-0	41.86	18.63	28.47	24.48	24.22	28.24	45.25	**27.33**	32.35	30.02	30.92	34.50
Vanilla-1	39.95	14.33	25.38	12.20	14.49	22.92	42.45	20.67	31.25	14.01	17.22	26.78
Vanilla-3	45.44	16.67	29.17	10.85	19.84	27.32	49.85	20.33	31.47	19.40	22.15	30.78
Vanilla-5	46.20	18.67	30.23	21.66	22.40	29.09	**53.38**	22.33	35.87	24.20	26.82	34.18
Vanilla-10	**51.32**	18.33	32.42	**27.32**	26.77	**32.14**	50.92	23.67	39.04	29.60	31.19	35.26
Rerank-1	41.38	16.33	**39.30**	16.40	20.38	26.03	44.55	22.33	39.94	18.01	22.06	29.65
Rerank-3	49.55	15.33	36.28	21.80	24.61	29.83	50.67	21.33	38.29	24.60	27.39	33.13
Rerank-5	48.78	15.67	35.35	22.81	25.39	29.95	49.30	21.67	38.11	28.23	30.28	33.75
Rerank-10	47.02	17.67	36.27	26.82	**28.22**	30.97	48.63	21.67	36.17	29.81	30.54	33.62
AdaRankGPT4	46.97	**19.03**	36.65	25.68	27.15	31.04	51.97	22.67	**43.05**	31.80	33.67	**36.10**
AdaRankLLM	45.27	17.02	36.25	25.59	27.25	29.84	49.05	24.04	41.02	**32.60**	**33.74**	35.60
Oracle	66.80	31.05	-	-	34.81	44.22	64.20	37.00	-	-	46.07	49.09
	Llama3.1-8B-Instruct						**Qwen2.5-7B-Instruct**					
Vanilla-0	35.49	**18.37**	16.98	23.10	16.86	23.57	27.02	**20.50**	15.24	10.06	11.05	19.52
Vanilla-1	32.93	11.83	22.63	12.00	13.69	19.48	35.83	14.90	24.06	10.70	13.18	21.30
Vanilla-3	41.30	13.50	24.63	18.38	18.70	24.50	43.91	17.13	26.59	15.88	17.74	26.26
Vanilla-5	44.75	15.13	25.45	23.56	21.77	27.22	45.58	17.00	28.72	18.46	19.85	27.48
Vanilla-10	48.31	17.10	25.73	28.56	23.73	29.71	49.05	18.80	29.08	21.86	22.15	30.00
Rerank-1	38.28	13.34	**30.52**	15.94	18.23	23.28	42.49	16.47	**33.08**	14.26	17.53	25.50
Rerank-3	44.57	15.30	30.06	23.54	23.51	27.79	47.25	17.50	31.59	19.20	21.06	28.60
Rerank-5	47.04	15.67	28.41	26.52	24.41	29.04	**49.16**	18.20	29.69	20.98	21.88	29.75
Rerank-10	**48.92**	17.20	25.66	**29.42**	24.02	**30.05**	49.05	18.47	30.31	22.16	22.58	30.03
AdaRankGPT4	45.88	16.50	30.02	28.22	**25.54**	29.31	47.84	19.77	31.19	**22.30**	**22.84**	**30.15**
AdaRankLLM	47.44	17.50	27.31	28.04	24.29	29.74	47.40	18.60	29.81	21.76	22.26	29.42
Oracle	63.32	30.47	-	-	37.52	43.77	60.50	32.33	-	-	31.93	41.58
	Qwen3-8B-NoThinking						**Qwen3-8B-Thinking**					
Vanilla-0	26.79	**20.67**	14.32	7.98	13.65	20.37	28.93	**23.27**	13.69	8.05	13.37	21.85
Vanilla-1	39.06	16.37	20.81	8.23	15.13	23.52	41.51	16.10	22.54	7.99	14.51	24.04
Vanilla-3	47.51	17.93	23.79	11.36	19.94	28.46	48.93	17.60	25.91	12.12	20.49	29.01
Vanilla-5	49.24	18.63	25.02	13.12	22.51	30.13	51.59	18.37	27.49	14.55	23.31	31.09
Vanilla-10	**52.77**	20.17	26.30	15.78	25.21	**32.72**	**54.85**	21.03	28.50	**17.61**	**26.66**	34.18
Rerank-1	44.42	17.57	27.37	10.08	18.99	26.99	46.00	17.93	**31.05**	10.46	18.95	27.63
Rerank-3	48.97	18.53	**29.04**	14.50	25.08	30.86	50.63	19.57	30.36	15.28	24.85	31.68
Rerank-5	50.92	19.73	28.27	15.42	25.74	32.13	52.64	19.10	29.22	16.57	25.81	32.52
Rerank-10	51.57	20.33	26.68	16.27	25.51	32.47	54.82	22.00	28.91	17.58	26.48	**34.43**
AdaRankGPT4	50.34	19.77	28.24	16.30	**26.23**	32.11	52.17	21.90	28.66	17.54	26.20	33.42
AdaRankLLM	50.40	19.93	28.16	**16.40**	26.09	32.14	51.00	21.50	28.52	17.01	25.76	32.75
Oracle	62.69	32.29	-	-	36.43	43.81	63.99	33.99	-	-	37.52	45.17

V. Results

A. main result

Table II presents the comparative performance across diverse models. Beyond demonstrating universal efficacy, the results underscore a critical **functional shift** in adaptive retrieval relative to model capability. While AdaRankLLM consistently approximates the optimal static strategies for each architecture, we observe that its primary utility evolves fundamentally: transitioning from an essential *noise filter* for weaker models to a strategic *efficiency optimizer* for stronger ones.

For weaker models, AdaRankLLM acts as a critical noise filter. Models with limited parameter counts or reasoning capabilities (e.g., Alpaca-7b, Mistral-7b) exhibit low noise robustness, causing their performance to degrade when exposed to larger retrieval contexts (Vanilla-10 / Rerank-10). For these models, the external purification provided by our adaptive ranking is essential. AdaRankLLM effectively discards irrelevant passages that would otherwise distract the model, matching the high precision of aggressive filtering strategies while maintaining the flexibility to recall more information when necessary.

For stronger models, AdaRankLLM serves as an efficiency optimizer. Advanced models, particularly those with explicit reasoning paths (e.g., Qwen3), possess strong internal attention mechanisms capable of handling noisy contexts (Vanilla-10) to maximize recall. However, this robustness comes at the cost of processing excessive irrelevant tokens. In these cases, AdaRankLLM maintains competitive performance—often indistinguishable from full-context retrieval—but drastically reduces the input length. This demonstrates that processing the full candidate set is often redundant; AdaRankLLM effectively extracts the "minimal sufficient context", eliminating the computational overhead of irrelevant tokens while preserving the critical evidence required for reasoning.

B. Further Analysis

a) ***The Fallacy of Static Retrieval.****:* Our analysis demonstrates that the optimal retrieval depth (k) is highly volatile, shifting significantly depending on the specific task, dataset, and backbone model. This variance renders any universal fixed-k configuration suboptimal—either risking information loss or incurring unnecessary computational costs. AdaRankLLM addresses this inherent unpredictability by dynamically calibrating context size for each query. It effectively eliminates the need for manual tuning, serving as an adaptive solution that autonomously navigates the trade-off between sufficiency and efficiency across diverse experimental setups.

b) ***Efficacy of Instruction Distillation.****:* To validate our distillation paradigm, we compare the performance of our distilled model directly against the teacher model, AdaRank-GPT4. Results indicate that the student model achieves performance parity with GPT-4 across evaluations, effectively mastering the complex criteria for listwise ranking and adaptive passage dropout. This confirms that the high-cost reasoning capabilities of proprietary models can be successfully compressed into smaller, open-source architectures, making advanced adaptive retrieval viable for efficient local deployment.

c) ***Noise Sensitivity of Weaker Architectures.****:* Weaker backbones (e.g., Alpaca, Mistral) exhibit high noise sensitivity, consistently achieving peak performance at minimal retrieval depths ($k = 1, 3$). Constrained by limited reasoning capabilities and effective context windows, these models degrade rapidly when exposed to larger, noisier retrieval sets. Consequently, they necessitate aggressive noise filtration rather than broad ingestion. AdaRankLLM excels in this scenario by effectively "purifying"" the context, automatically balancing precision and recall to align with the model's restricted processing capacity.

d) ***Noise Robustness and Internal Verification in Advanced Models****:* In stark contrast to weaker counterparts, advanced models favor maximum information intake, achieving peak performance under high-recall settings (e.g., Vanilla-10). This indicates that for capable architectures, *information recall* takes precedence over precision. Utilizing sophisticated attention mechanisms, they isolate valid evidence from noise, meaning they suffer more from missing information due to aggressive filtering than from the presence of irrelevant data. This robustness is amplified in models with explicit *Thinking* modes. We observe that Vanilla-10 often yields the best results here, suggesting that explicit reasoning [40] acts as a potent **internal verification mechanism**. By logically distinguishing conflicting data step-by-step, these models "internalize" the noise-filtering step, enabling them to exploit the full breadth of raw retrieval sets without being misled by inherent noise.

e) ***Performance Gap to Theoretical Optima****:* Finally, across all evaluated architectures, a persistent disparity remains between the best achieved performance and the theoretical upper bound represented by the idealized "Oracle". This observation underscores that the fundamental challenges of Retrieval-Augmented Generation remain far from being fully resolved. The persistence of this gap suggests that strategies relying solely on increasing model scale, expanding context windows, or employing rudimentary adaptive filtering are beginning to yield diminishing marginal returns, highlighting the critical need for more sophisticated retrieval-integration paradigms to effectively bridge this disparity.

VI. Conclusion

In this paper, we propose AdaRankLLM, a lightweight framework that endows smaller models with adaptive ranking capabilities via instruction distillation. Our experiments reveal that the primary utility of adaptive retrieval shifts with model capability: it functions as a critical noise filter for weaker models and an efficiency optimizer for stronger ones. Furthermore, the persistent gap between current state-of-the-art results and the theoretical "Oracle"" indicates that despite the continuous advancement of model capabilities, the fundamental challenge of optimal information retrieval remains unresolved. Future work will investigate agentic paradigms to enable adaptive retrieval to approximate the theoretical optimal solution.

VII. Acknowledgement

This work was supported by the National Natural Science Foundation of China (Nos. U23A20319 and 62441239), as well as the Anhui Province Science and Technology Inno vation Project (Nos. 202423k09020010 and 202423k09020011).

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[2] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan, "Query rewriting in retrieval-augmented large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 5303–5315. [Online]. Available: https://aclanthology.org/2023.emnlp-main.322

[3] H. Lv, S. Liang, H. Wang, H. Gu, Y. Wu, W. Guo, D. Lian, Y. Liu, and E. Chen, "Costeer: Collaborative decoding-time personalization via local delta steering," 2026. [Online]. Available: https://arxiv.org/abs/2507.04756

[4] Y. Huang, R. Zhang, Q. Wang, C. Lu, Y. Gao, Y. Wu, Y. Hu, X. Zhi, G. Liu, X. Li, H. Wang, and E. Chen, "Selfaug: Mitigating catastrophic forgetting in retrieval-augmented generation via distribution self-alignment," 2025. [Online]. Available: https://arxiv.org/abs/2509.03934

[5] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen, L. Wang, A. T. Luu, W. Bi, F. Shi, and S. Shi, "Siren's song in the ai ocean: A survey on hallucination in large language models," 2023.

[6] Z. Ke, Y. Shao, H. Lin, T. Konishi, G. Kim, and B. Liu, "Continual pre-training of language models," in *The Eleventh International Conference on Learning Representations*, 2022.

[7] R. Ren, Y. Wang, Y. Qu, W. X. Zhao, J. Liu, H. Tian, H. Wu, J.-R. Wen, and H. Wang, "Investigating the factual knowledge boundary of large language models with retrieval augmentation," 2023.

[8] F. Shi, X. Chen, K. Misra, N. Scales, D. Dohan, E. H. Chi, N. Schärli, and D. Zhou, "Large language models can be easily distracted by irrelevant context," in *International Conference on Machine Learning*. PMLR, 2023, pp. 31 210–31 227.

[9] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-rag: Learning to retrieve, generate, and critique through self-reflection," *arXiv preprint arXiv:2310.11511*, 2023.

[10] Y. Tu, W. Su, Y. Zhou, Y. Liu, and Q. Ai, "Rbft: Robust fine-tuning for retrieval-augmented generation against retrieval defects," 2025. [Online]. Available: https://arxiv.org/abs/2501.18365

[11] D. S. Sachan, M. Lewis, M. Joshi, A. Aghajanyan, W.-t. Yih, J. Pineau, and L. Zettlemoyer, "Improving passage retrieval with zero-shot question generation," *arXiv preprint arXiv:2204.07496*, 2022.

[12] Z. Qin, R. Jagerman, K. Hui, H. Zhuang, J. Wu, J. Shen, T. Liu, J. Liu, D. Metzler, X. Wang *et al.*, "Large language models are effective text rankers with pairwise ranking prompting," *arXiv preprint arXiv:2306.17563*, 2023.

[13] X. Ma, X. Zhang, R. Pradeep, and J. Lin, "Zero-shot listwise document reranking with a large language model," 2023.

[14] Q. Liu, B. Wang, N. Wang, and J. Mao, "Leveraging passage embeddings for efficient listwise reranking with large language models," in *THE WEB CONFERENCE 2025*, 2024.

[15] H. Lv, H. Gu, R. Yang, L. Li, Z. Chen, D. Lian, H. Wang, and E. Chen, "Learning from emptiness: De-biasing listwise rerankers with content-agnostic probability calibration," 2026. [Online]. Available: https://arxiv.org/abs/2604.10150

[16] R. Pradeep, S. Sharifymoghaddam, and J. Lin, "Rankvicuna: Zero-shot listwise document reranking with open-source large language models," *arXiv preprint arXiv:2309.15088*, 2023.

[17] ——, "Rankzephyr: Effective and robust zero-shot listwise reranking is a breeze!" *arXiv preprint arXiv:2312.02724*, 2023.

[18] M. S. Tamber, R. Pradeep, and J. Lin, "Scaling down, litting up: Efficient zero-shot listwise reranking with seq2seq encoder-decoder models," *arXiv preprint arXiv:2312.16098*, 2023.

[19] S. Zhuang, X. Ma, B. Koopman, J. Lin, and G. Zuccon, "Rank-r1: Enhancing reasoning in llm-based document rerankers via reinforcement learning," *arXiv preprint arXiv:2503.06034*, 2025.

[20] Q. Liu, B. Wang, N. Wang, and J. Mao, "Leveraging passage embeddings for efficient listwise reranking with large language models," in *Proceedings of the ACM on Web Conference 2025*, 2025, pp. 4274–4283.

[21] S. Chen, B. J. Gutiérrez, and Y. Su, "Attention in large language models yields efficient zero-shot re-rankers," *arXiv preprint arXiv:2410.02642*, 2024.

[22] S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. C. Park, "Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity," *arXiv preprint arXiv:2403.14403*, 2024.

[23] H. Liu, H. Zhang, Z. Guo, K. Dong, X. Li, Y. Q. Lee, C. Zhang, and Y. Liu, "Ctrla: Adaptive retrieval-augmented generation via probe-guided control," *arXiv preprint arXiv:2405.18727*, 2024.

[24] Z. Jiang, F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, and G. Neubig, "Active retrieval augmented generation," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 7969–7992. [Online]. Available: https://aclanthology.org/2023.emnlp-main.495

[25] H. Liu, H. Zhang, Z. Guo, J. Wang, K. Dong, X. Li, Y. Q. Lee, C. Zhang, and Y. Liu, "Ctrla: Adaptive retrieval-augmented generation via inherent control," *arXiv preprint arXiv:2405.18727*, 2024.

[26] Y. Wang, H. Ling *et al.*, "Targ: Training-free adaptive retrieval gating for efficient rag," *arXiv preprint arXiv:2511.09803*, 2025.

[27] M. A. Asl, M. Asgari-Bidhendi, and B. Minaei-Bidgoli, "Fair-rag: Faithful adaptive iterative refinement for retrieval-augmented generation," *arXiv preprint arXiv:2510.22344*, 2025.

[28] J. Kim, J. Kim, S.-w. Hwang, J. Kim, Y. J. Kim, and M. Lee, "Adaptive retrieval for reasoning-intensive retrieval," *arXiv preprint arXiv:2601.04618*, 2026.

[29] H. Gu, D. Li, K. Dong, H. Zhang, H. Lv, H. Wang, D. Lian, Y. Liu, and E. Chen, "Rapid: Efficient retrieval-augmented long text generation with writing planning and information discovery," *arXiv preprint arXiv:2503.00751*, 2025.

[30] H. Hashemi, V. Rühle, and S. Rajmohan, "Cost-aware retrieval-augmentation reasoning models with adaptive retrieval depth," *arXiv preprint arXiv:2510.15719*, 2025.

[31] I. Stelmakh, Y. Luan, B. Dhingra, and M.-W. Chang, "Asqa: Factoid questions meet long-form answers," *arXiv preprint arXiv:2204.06092*, 2022.

[32] S. Amouyal, T. Wolfson, O. Rubin, O. Yoran, J. Herzig, and J. Berant, "Qampari: A benchmark for open-domain questions with many answers," in *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

[33] A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli, "Eli5: Long form question answering," *arXiv preprint arXiv:1907.09190*, 2019.

[34] T. Gao, H. Yen, J. Yu, and D. Chen, "Enabling large language models to generate text with citations," *arXiv preprint arXiv:2305.14627*, 2023.

[35] W. Sun, L. Yan, X. Ma, P. Ren, D. Yin, and Z. Ren, "Is chatgpt good at search? investigating large language models as re-ranking agent," *arXiv preprint arXiv:2304.09542*, 2023.

[36] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[37] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, "Ms marco: A human generated machine reading comprehension dataset," *choice*, vol. 2640, p. 660, 2016.

[38] A. Neelakantan, T. Xu, R. Puri, A. Radford, J. M. Han, J. Tworek, Q. Yuan, N. Tezak, J. W. Kim, C. Hallacy *et al.*, "Text and code embeddings by contrastive pre-training," *arXiv preprint arXiv:2201.10005*, 2022.

[39] O. Honovich, R. Aharoni, J. Herzig, H. Taitelbaum, D. Kukliansy, V. Cohen, T. Scialom, I. Szpektor, A. Hassidim, and Y. Matias, "TRUE: Re-evaluating factual consistency evaluation," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 3905–3920. [Online]. Available: https://aclanthology.org/2022.naacl-main.287

[40] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.