RetroLM: Retrieval-Augmented KVs for Long-Context Processing

Kun Luo[1,2,3], Zheng Liu[2,4*], Shitao Xiao[2], Jiabei Chen[1,2,3], Hongjin Qian[2,5], Peitian Zhang[2], Shanshan Jiang[6], Bin Dong[6], Jun Zhao[1,3], Kang Liu[1,3*]

[1]The Key Laboratory of Cognition and Decision Intelligence for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]Beijing Academy of Artificial Intelligence, Beijing, China
[3]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[4]Hong Kong Polytechnic University, Hong Kong, China
[5]Peking University, Beijing, China,
[6]Ricoh Software Research Center Beijing, Ricoh Company, Ltd.
{luokun695, zhengliu1026}@gmail.com, kliu@nlpr.ia.ac.cn

Abstract

Long-context processing remains a significant challenge for large language models (LLMs). Retrieval-augmented generation (RAG) has recently emerged as a promising approach, enabling LLMs to selectively access relevant information from extended contexts to improve efficiency. However, existing RAG approaches often lag behind other efficient long-context processing methods primarily due to inherent limitations on inaccurate retrieval and fragmented contexts. To address these limitations, we propose **RetroLM**, a novel RAG framework designed for effective long-context processing. Unlike traditional approaches, RetroLM introduces **KV-level retrieval augmentation**, which partitions the LLM's Key-Value (KV) cache into contiguous pages and performs encoding and decoding operations based on the retrieved KV pages. Built upon this framework, we further develop a **specialized retriever** for precise retrieval of critical pages and conduct **unsupervised post-training** to optimize the model's ability to leverage retrieved information. Compared with traditional RAG, the new approach enhances robustness to retrieval inaccuracy, facilitates effective utilization of fragmented contexts, and saves the cost from repeated context-encoding operations. We conduct extensive evaluations across several popular benchmarks, including LongBench, InfiniteBench, and RULER. RetroLM consistently outperforms existing long-LLMs and RAG-based methods, especially in tasks requiring deep reasoning or extreme context lengths.

Code — https://github.com/kunlun531/RetroLM

Introduction

The processing of long contexts has emerged as a critical issue in the development and application of Large Language Models (LLMs). Numerous applications necessitate the ability to handle extended sequences of information, including understanding lengthy documents (Bai et al. 2023; Caciularu et al. 2023), supporting sophisticated AI agent systems (Jin et al. 2024), and generating long-form reasoning chains for complex tasks, such as mathematical proofs (OpenAI 2024) or computer programming (Gur et al. 2023). To address this crucial requirement, substantial efforts have been devoted to extending the maximum context lengths accommodated by LLMs. For example, GPT-4 (Achiam et al. 2023) and LLaMA-3.1 (Dubey et al. 2024), both of which support a 128K token context window. Moreover, the recent Gemini2.5-Pro (Comanici et al. 2025) makes a dramatic extension, enabling a context window of over 10M tokens.

Despite these impressive naive extensions, the practical application of long-context models is severely limited by a fundamental bottleneck: the computational and memory cost of self-attention. The quadratic complexity of attention computation and the linear growth of the Key-Value (KV) cache with sequence length render naive full-attention prohibitively expensive. This has spurred the development of two main classes of efficient processing methods, each with its own inherent limitations. The first class, heuristic-based KV compression, includes methods like StreamingLLM (Xiao et al. 2023a), H2O (Zhang et al. 2023b), and SnapKV (Li et al. 2024b). These approaches rely on heuristics (such as token recency or attention scores) as proxies for importance to decide which KVs to discard. However, such heuristics are often insufficient for tasks requiring complex, non-local reasoning, leading to a performance ceiling as critical information may be prematurely dropped. The second class, trainable sparse attention (Lu et al. 2025; Yuan et al. 2025), trains models from scratch to adapt to sparse KV caches. While potentially more performant, this approach requires massive, costly pre-training, making it impractical for adapting existing state-of-the-art LLMs.

As an alternative paradigm, Retrieval-Augmented Generation (RAG) has shown promise by retrieving relevant information from external knowledge sources. Inspired by this, recent studies apply RAG to the long-context problem by chunking long documents, retrieving the most relevant chunks, and feeding them to the LLM (Xu et al. 2023; Li et al. 2024a). However, this text-level RAG paradigm faces two inherent limitations. First, it causes semantic fragmentation: the hard division of text into chunks shatters the document's narrative and structural coherence, preventing LLM from grasping global context and inter-segment rela-

*Corresponding authors.

tionships (Luo et al. 2024; Li et al. 2024a). Second, it creates a retrieval-generation gap: the retriever and the LLM are disparate systems. A retrieval error is a "hard" failure; if a relevant chunk is missed, the LLM permanently loses access to that information, making the entire system overly sensitive to retriever accuracy (Bai et al. 2023; Qian et al. 2024).

In this work, we introduce **RetroLM**, a novel framework that performs retrieval augmentation directly at the **KV cache level** to address these challenges. Instead of manipulating raw text, RetroLM partitions the LLM's KV cache into contiguous pages. During inference, it retrieves only the most crucial pages for the attention computation. This design offers distinct advantages over traditional RAG (Zhang et al. 2023a). By operating on the KV cache, it is more robust to retrieval inaccuracies and can naturally handle fragmented information through the inherent sparsity of the attention mechanism (Jiang et al. 2024).

Compared to existing sparse attention and KV compression methods, RetroLM introduces a dedicated, trainable **page retriever** that learns to identify important KV pages through fine-grained interactions, rather than relying on fixed heuristics (Zhang et al. 2023b; Li et al. 2024b). Furthermore, RetroLM circumvents the need for costly native sparse attention pre-training from scratch (Lu et al. 2025). We propose an efficient two-stage fine-tuning process. Stage 1 focuses on training the page retriever, requiring only minimal fine-tuning on curated datasets; Stage 2 adapts the full model to sparsely retrieved contexts using unlabeled data with a constrained training length (12K). This approach significantly enhances performance and generalization across diverse tasks and context lengths.

We perform comprehensive evaluations using several standard benchmarks in this field, including LongBench (Bai et al. 2023), InfiniteBench (Zhang et al. 2024), and RULER (Hsieh et al. 2024). In our experiment, RetroLM outperforms popular efficient long-context processing methods with notable advantages. In majority of the tasks, it achieves an equivalent performance as the expensive full-attention methods; while for certain scenarios like long-doc QA, it even surpasses full-attention by effectively filtering out background.

Related Work

In this section, we make discussions on the following related works: 1) context extension of LLMs, 2) efficient long-context processing, 3) RAG for long-context processing.

First of all, a substantial body of research has focused on extending the context length of LLMs directly. One common approach involves modifying positional encoding mechanisms to enable LLMs trained on short texts to process longer inputs directly during inference (Chen et al. 2023a; Peng et al. 2023; Ding et al. 2024). While straightforward, these methods often yield suboptimal performance without additional fine-tuning. Another widely adopted strategy is continual training, where existing LLMs are fine-tuned on long-sequence data to expand their context windows (Li et al. 2023; Chen et al. 2023b; Mohtashami and Jaggi 2023). However, fine-tuning approaches typically require training from extremely long-sequence data, which is challenging due to the scarcity of native human-annotation data and the high expenses resulted from the training operations (Fu et al. 2024; Gao et al. 2024).

Among efficient long-context processing methods, sparse attention and KV cache sparsification has gained significant attention for their ability to selectively utilize portions of KVs based on certain reduction strategies, where KVs are reduced into a fixed budget (e.g., 2K) (Xu et al. 2024; Tang et al. 2024; Liu et al. 2024; Zhang et al. 2023b). For instance, InfLLM (Xiao et al. 2024) incorporates intermediate information by segmenting KVs into fixed-size chunks and selecting top-k most salient chunks based on attention score patterns. SnapKV and PyramidKV (Li et al. 2024b; Cai et al. 2024) extend to alleviate memory pressure during the prefilling stage by dropping tokens based on cumulative attention scores within localized windows. However, these heuristic-based methods are fundamentally limited by their reliance on full attention score estimation, which restricts their effectiveness in complex long-context tasks. To address these limitations, we propose a specialized, trainable KV page retriever which is well-suited for handling complex, long-context tasks.

More recent studies proposed native trainable sparse attention mechanisms (Lu et al. 2025; Yuan et al. 2025), which allow LLMs to adapt to sparse KV from pre-training phase. While promising, these approaches incur substantial computational costs due to the need for large-scale pretraining with sparse attention objectives. In contrast, RetroLM achieves effective long-context adaptation with significantly lower overhead. In the first stage, it requires only minimal fine-tuning on curated datasets with explicit supervision to train the KV page retriever. In the second stage, a few hours of additional training on unsupervised text allow RetroLM to adapt to sparse KV settings and long-context inputs. This two-stage design prioritizes both computational efficiency and generalization across diverse tasks and context lengths, all while preserving the original capabilities of the base LLM.

Retrieval-augmented generation (RAG) has emerged as a promising approach for addressing long-context tasks (Xu et al. 2023; Li et al. 2024a; Yue et al. 2024). Leveraging modern dense retrievers (Karpukhin et al. 2020; Xiao et al. 2023b), these approaches first partition the long text into smaller chunks, subsequently selecting the most salient chunks, and concatenating them to form a new prompt for the LLM (Zhao et al. 2024). In addition, several specialized retrievers have been developed for long-context scenarios (Luo et al. 2024; Günther et al. 2023). In this work, RetroLM integrates retrieval augmentation directly at the KV cache level, thereby seamlessly incorporating RAG pipeline into long-context language modeling.

Method

Problem Formulation

For long-context understanding and language modeling tasks, such as question answering, summarization, the input can be structured into: context X, user query q, and output

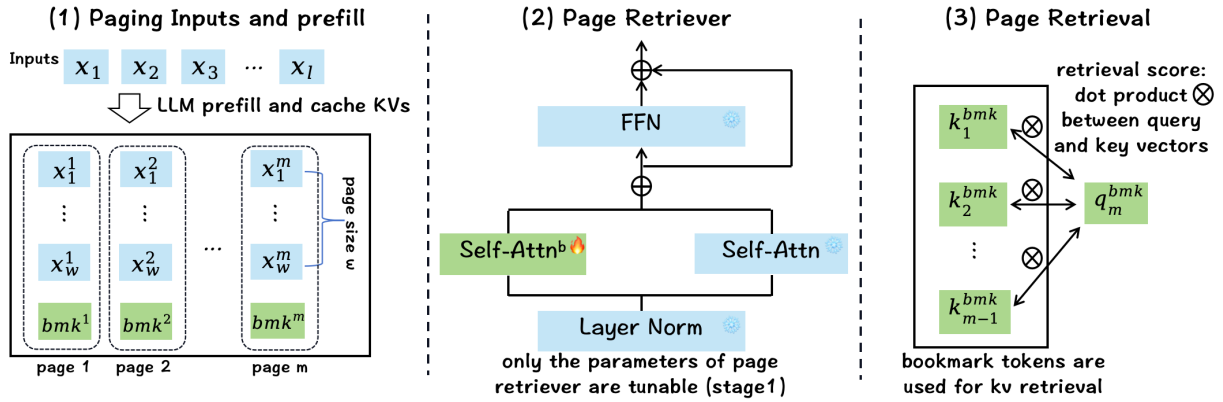

Figure 1: Framework of RetroLM: *(1)* Paging mechanism for KV management. *(2)* Specialized trainable, plug-in page retriever. *(3)* Page retrieval using special bookmark tokens, with their representation established within attention module.

Y. The generation objective of LLM can be expressed as:

$$max. \log \text{LLM}(y_t|X, q, Y_{<t}) \quad (1)$$

In such scenarios, the context X often exceeds 100K tokens, leading to significant computational and memory consumption. To address this problem, various efficient long-context processing techniques have been introduced (Xiao et al. 2023a, 2024; Zhang et al. 2023b), aiming at compressing context either implicitly or explicitly using a designated reduction policy $p(X)$.

RAG-based methods employ a standalone retriever as an explicit context reduction policy $p^{ret}(\cdot)$ (Chen et al. 2024; Jiang, Ma, and Chen 2024; Zhang et al. 2023a). It first chunks the long context into: $X : \{s_1, ..., s_N\}$, and then select the top-k relevant chunks: $X^{ret} : \{s_1, ..., s_k\}$. The X^{ret} forms the new input context. Explicit context compression of RAG-based methods prune the prompt rigidly, which results in information loss and semantic discontinuities.

RetroLM performs retrieval augmentation at KV cache level, using a plug-in page retriever as policy $p^{kv}(\cdot)$. It selects the most crucial KVs at each decoder layer: $C = p^{kv}(X)$, where C is the KVs for attention computation, thereby achieving implicit context compression. Unlike existing KV sparsification approaches that rely on heuristic or unsupervised methods to approximate full attention, RetroLM introduces a specialized and trainable page retriever, inspired by dense retrieval techniques. Further analysis is conducted in case study section.

Inference Process

Paging Inputs. RetroLM first partitions the LLM's input context $X = \{x_i\}_{i=1}^{l}$ into contiguous pages:

$$\{x_1, ..., x_l\} \xrightarrow{\text{partition}} \{X_1, ..., X_m\}, X_i = \{x_j^i\}_{j=1}^{w} \quad (2)$$

where w is the page size (128 in practice), l is the sequence length, and m is the number of pages ($m = \lceil \frac{l}{w} \rceil$). Then for each page X_i, a special bookmark token ($\langle \text{BMK} \rangle$) is inserted to the end of it: $X_i^{'} = \{x_1^i, ..., x_w^i, \langle bmk \rangle^i\}$. The LLM encodes both the normal tokens and bookmark tokens. The bookmark tokens function as the ***page indexes*** of corresponding pages for KV retrieval and establish their representations during attention computation across each decoder layer.

Pre-filling. During pre-filling, we employ streaming encoding based on page retrieval to enable the process of extremely long inputs. Specifically, a fixed-sized sliding window is used to encode the long context progressively. In each layer, the encoding of page $X_i{}'$ only retrieves k pages (including the first page as attention sink) for attention computation instead of costly full attention:

$$C : \{X_1', ..., X_k'\} = p^{kv}(X' : \{X_1', ..., X_{i-1}'\}|X_i') \quad (3)$$

Once encoded, the KVs of page X_i' are offloaded to CPU, ensuring that only the required KV pages are reloaded to GPU for attention computation.

Decoding. During decoding, page retrieval is conducted only once given the user query:

$$C : \{X_1', ..., X_k'\} = p^{kv}(X' : \{X_1', ..., X_m'\}|q) \quad (4)$$

Page Retriever

Architecture. This paper propose a trainable, plug-and-play page retriever designed to conduct KV cache level retrieval augmentation, whose architecture is shown in Figure 1 (Middle). It reuses all modules of the LLM except imposing a slight modification on the self-attention module.

During the self-attention computation, the hidden states of normal tokens (n) and bookmark tokens (b) are sliced out and projected into query, key, and value vectors respectively:

$$\begin{aligned} \boldsymbol{Q}^n &= \boldsymbol{W}_Q^n \boldsymbol{H}^n, \quad \boldsymbol{K}^n = \boldsymbol{W}_K^n \boldsymbol{H}^n, \quad \boldsymbol{V}^n = \boldsymbol{W}_V^n \boldsymbol{H}^n, \\ \boldsymbol{Q}^b &= \boldsymbol{W}_Q^b \boldsymbol{H}^b, \quad \boldsymbol{K}^b = \boldsymbol{W}_K^b \boldsymbol{H}^b, \quad \boldsymbol{V}^b = \boldsymbol{W}_V^b \boldsymbol{H}^b \end{aligned} \quad (5)$$

where $\boldsymbol{W}_*^{\boldsymbol{n}}$ are the LLM's original projection matrices and $\boldsymbol{W}_*^{\boldsymbol{b}}$ are the newly introduced matrices designed specifically to handle bookmark tokens. The bookmark tokens distill corresponding page's contextual information during attention computation and are used for page retrieval.

Retrieval Score. Page importance estimation employs similarity between the query vector of target page's bookmark token and the key vectors of past pages' bookmark tokens:

$$p^{kv}(\{X_1', ..., X_{m-1}'\}|X_m') = \text{top-}k \left\{ \langle \boldsymbol{q}_m^{bmk}, \boldsymbol{k}_j^{bmk} \rangle \right\}_{j=1}^{m-1} \quad (6)$$

where $\langle *, * \rangle$ denotes the dot product operation, commonly used as a similarity measurement in dense retrieval (Karpukhin et al. 2020).

Two-Stage Training Framework for RetroLM

To effectively equip RetroLM for long-context processing, we introduce a cohesive two-stage training framework. This design strategically decouples the task into two subproblems: first, learning a precise retrieval mechanism for KV pages, and second, adapting the language model to effectively utilize these sparsely retrieved contexts. This design not only enhances performance but also offers significant flexibility in plug-and-play deployment.

Stage-1: Training the Page Retriever via Contrastive Learning. The first phase, **Stage-1**, focuses exclusively on training the page retriever to identify salient KV pages within complex and distracting contexts. During this stage, all parameters of the backbone LLM are kept frozen, allowing us to efficiently instill specialized retrieval capabilities into the lightweight page retriever module.

The primary challenge is the lack of direct supervision signals for KV page importance. To overcome this, we adopt a contrastive learning approach, inspired by its success in dense retrieval (Karpukhin et al. 2020; Chen et al. 2024; Luo et al. 2024). The objective is to train the retriever to distinguish a single positive page containing relevant information from a set of hard-negative distractor pages. Assuming the query resides on page m and the relevant information is on page i, the contrastive objective is defined as:

$$L_1 = -\log \frac{\exp(\langle \boldsymbol{q}_m^{\text{bmk}}, \boldsymbol{k}_i^{\text{bmk}} \rangle)}{\sum_{j=1}^{m-1} \exp(\langle \boldsymbol{q}_m^{\text{bmk}}, \boldsymbol{k}_j^{\text{bmk}} \rangle)} \tag{7}$$

where $\boldsymbol{q}^{\text{bmk}}$ and $\boldsymbol{k}^{\text{bmk}}$ are the query and key vectors derived from the special bookmark tokens. To construct robust training instances, we leverage 50K web search examples from MS MARCO (Bajaj et al. 2016) and synthesize an additional 5K coherent question-answer pairs from Slimpajama (Shen et al. 2023). The detailed data curation is described in the *Appendix Data Formation*. Upon completion, Stage-1 yields a highly proficient page retriever, ready to guide the attention mechanism in the subsequent full-model adaptation phase.

Stage-2: Full-Model Adaptation to Retrieved KV Caches. Following the specialized training of the page retriever, we proceed to **Stage-2**: a post-training phase designed to adapt the entire RetroLM model to operate with the retrieved sparse KV caches. The key objective here is to teach the backbone LLM to effectively process and reason over the sparse and non-contiguous KVs supplied by the Stage-1 retriever.

In this stage, we unfreeze the backbone LLM and perform a short, efficient fine-tuning session (approx. 5 hours) using unsupervised data from Slimpajama (Shen et al. 2023) with a maximum length of 12K tokens. During this process, we simulate the exact inference-time behavior: the well-trained page retriever from Stage-1 dynamically selects a budget of top-k pages to form a sparse KV cache, C^{ret}. The model is then updated using a standard causal language modeling objective on this sparse context:

$$L_2 = -\sum_t \log P(x_t | x_{<t}, C^{\text{ret}}) \tag{8}$$

Notably, the goal of Stage-2 is not length extension but **adaptation to sparsity**. By fine-tuning the model to handle the sparse KVs it will encounter during inference, we enhance its capacity to leverage the retrieved pages effectively. This two-stage approach allows us to first master precise selection (Stage-1) and then optimize the model's ability to utilize these selections for coherent generation (Stage-2), forming the core of RetroLM's efficiency and effectiveness.

Experiment

We conduct extensive experiments focused on answering the following two research questions: 1) The effectiveness of RetroLM against long-context LLMs and other efficient methods in long-context understanding tasks. 2) How well can RetroLM generalize to different long-context tasks and context lengths.

Experimental Setting

Datasets. To comprehensively evaluate the overall performance of RetroLM, we employ the **LongBench** (Bai et al. 2023). This benchmark encompasses a variety of long-context tasks, including **single-doc QA**, **multi-hop QA**, **Long doc summarization**, and **long ICL**. These tasks are well-suited for assessing the long-context capability in practical application scenarios. Subsequently, to assess the generalization of RetroLM in extremely long scenarios, we utilize tasks from **InfiniteBench** (Zhang et al. 2024), including free-form QA on long books (QA), summarization over long texts (Summary), multiple-choice QA on long books (Choice), and finding special numbers in lengthy lists (Math.F). The average input length within InfiniteBench is 145K tokens. We also use **RULER** (Hsieh et al. 2024) to evaluate long context key information identification capability. All evaluation metrics are aligned with official implementation.

Baseline Methods. To rigorously demonstrate the effectiveness of RetroLM, we compare its performance against the following competitive baseline methods: (1) Original Models: We report the performance of the LLMs with full attention mechanisms (Mistral-7B-Instruct and Llama-3-8B-Instruct) (Jiang et al. 2023; Dubey et al. 2024). (2) Stream Processing: This category includes methods like LM-Infinite (Han et al. 2023) and StreamingLLM (Xiao et al. 2023a), which employ attention sink and sliding window mechanisms for processing long inputs. (3) KV Compression: These methods, such as H2O (Zhang et al. 2023b), SnapKV (Li et al. 2024b), InfLLM (Xiao et al. 2024), and PyramidKV (Cai et al. 2024), employ heuristic KV sparsification policies to selectively retain portions of KVs. (4) RAG: We employ several retrieval methods to conduct RAG pipeline: the classic BM25 method (Robertson, Zaragoza et al. 2009), the Contriever model (Izacard et al. 2021), and the strong BGE-large-v1.5 model (Xiao et al. 2023b).

Effectiveness over other Efficient Methods on Diverse Long-context Understanding Tasks

To validate the effectiveness of RetroLM against other efficient methods in long-context understanding tasks, we con-

Model	Context	Narrative	Qasper	Multifield	Hotpot	2wikim	Musique	GovReport	MultiNews	QmSum	Trec	Trivia	SAMSum	Average
Mistral-7B-Instruct-v0.2														
Mistral-7B-v0.2	32k	**26.9**	33.1	49.2	43.0	27.3	18.8	25.6	26.2	23.3	**71.0**	86.2	42.6	39.4
LM-Infinite	2k	20.4	26.9	45.1	36.1	24.2	14.0	27.1	24.3	21.6	68.0	72.2	31.7	34.3
StreamingLLM	2k	20.3	26.6	45.7	35.3	24.3	12.2	27.5	24.5	21.6	68.5	71.9	31.2	34.1
InfLLM	2k	23.5	28.8	47.7	41.3	25.7	17.5	29.1	26.3	21.2	68.0	84.4	41.4	37.9
H2O	2k	25.6	31.1	49.0	40.8	26.5	17.1	24.8	26.6	23.6	55.0	86.3	42.4	37.4
SnapKV	2k	25.9	32.9	48.6	43.0	27.4	19.0	26.6	26.7	24.4	70.0	86.2	42.5	39.4
PyramidKV	2k	25.5	32.2	49.0	42.3	27.5	19.4	26.6	26.7	24.0	**71.0**	86.2	42.9	39.4
RetroLM-Stage1	2k	26.8	34.0	50.8	47.6	39.0	22.5	29.3	27.3	24.6	69.5	88.8	42.4	41.9
RetroLM-Stage2	2k	26.6	**38.7**	**53.8**	**47.7**	**41.6**	**26.4**	**29.8**	**28.2**	**25.9**	70.5	**89.3**	**43.0**	**43.5**
Llama-3-8B-Instruct														
Llama-3-8B	8k	25.8	29.6	41.0	45.4	36.1	22.9	26.2	26.5	23.4	74.0	90.5	42.3	40.3
LM-Infinite	2k	22.0	26.2	38.3	40.5	33.1	17.1	23.0	26.5	22.5	70.0	83.1	32.2	36.2
StreamingLLM	2k	21.7	25.8	38.1	40.1	32.0	16.9	23.1	26.5	22.6	70.0	83.2	31.8	36.0
InfLLM	2k	23.4	29.0	40.9	41.5	34.3	19.7	25.7	26.8	22.4	73.0	89.9	41.3	39.0
H2O	2k	25.6	26.9	39.5	44.3	32.9	21.1	24.7	24.6	23.0	53.0	90.5	41.8	37.3
SnapKV	2k	25.9	29.6	41.1	45.0	35.8	21.8	26.0	26.5	23.4	73.5	90.5	41.6	40.1
PyramidKV	2k	25.4	29.7	40.3	44.8	35.3	22.0	26.8	26.2	23.3	73.0	90.5	42.1	40.0
RetroLM-Stage1	2k	25.4	33.8	48.7	50.2	39.8	24.1	26.9	27.0	24.7	73.5	**91.0**	42.2	42.3
RetroLM-Stage2	2k	**26.6**	**38.7**	**48.9**	**52.5**	**45.4**	**27.0**	**30.4**	**27.9**	**26.1**	**75.5**	90.7	**42.8**	**44.4**

Table 1: Experiment results of comparing RetroLM with other efficient processing methods on LongBench. For each model section, the best results per column are in **bold**, second-best are underlined.

Model	Context	Narrative	Qasper	Multifield	Hotpot	2wikim	Musique	Average
Mistral-7B-v0.2	32k	**26.9**	33.1	49.2	43.0	27.3	18.8	33.1
Mistral-BM25	2k	13.9	22.7	34.6	31.0	22.7	17.8	23.8
Mistral-Contriever	2k	20.8	30.7	47.2	35.7	30.1	18.2	30.4
Mistral-BGE	2k	22.4	31.2	47.8	37.9	30.6	18.5	31.4
RetroLM-Stage1	2k	26.8	34.0	50.8	47.6	39.0	22.5	36.8
RetroLM-Stage2	2k	26.6	**38.7**	**53.8**	**47.7**	**41.6**	**26.4**	**39.1**

Table 2: Experiment results comparing RetroLM with RAG methods. The best results are in **bold**, second-best are underlined.

duct experiments on LongBench (Bai et al. 2023) with two popular backbone LLM (Mistral-7B-Instruct and Llama-3-8B-Instruct). The results are presented in Table 1. For the original models, we evaluate using their maximum context lengths. For RetroLM and other baseline methods, a fixed KV budget of 2K tokens is employed. Consequently, in each decoder layer's attention module, 2K tokens are selected for attention computation according to each method's respective KV reduction policy.

For the Mistral-based models, RetroLM achieves an overall score that surpasses all baselines, also significantly outperforming results obtained using full attention. Other approaches that employ heuristic KV selection strategies encounter performance ceilings comparable to full attention. Notably, RetroLM exceeds the performance of full attention by 2.5 points, even when only the KV retriever is trained during stage-1, with the language model remaining frozen. By learning to discriminate key information during the training of page retriever, RetroLM effectively identifies important KVs within extensive texts, achieving significant performance gains under constrained token budgets.

During stage-2 training of RetroLM, additional adaptation of LLM on unsupervised text data yields further performance improvements across tasks. This demonstrates the model's ability to adapt effectively to sparse KV cache and streaming encoding paradigm. To validate and analyze these findings, we conducted ablation studies using the same data but trained and evaluated the models with full attention. Similar trends are observed in experiments with the Llama-3-based models, corroborating the generality of our findings.

Advantages over RAG on Long-context Tasks

In this section, we compare RetroLM with traditional RAG methods (Robertson, Zaragoza et al. 2009; Izacard et al. 2021; Xiao et al. 2023b), which similarly aim to identify and utilize query-relevant information from long contexts. The experimental results on LongBench (Bai et al. 2023) QA tasks are presented in Table 2. For RAG, we retrieve the top 10 most salient chunks (each 200 tokens) for each dataset, resulting in a total budget of 2K tokens, matching the token budget used by RetroLM.

The results demonstrate that RetroLM consistently out-

Model	Context	QA	Summary	Choice	Math.F	Average
Mistral-7B-v0.2	32k	12.9	25.9	44.5	20.6	25.9
StreamingLLM	6k	10.9	21.0	40.4	15.1	21.8
H2O	6k	14.2	23.7	43.7	24.2	26.5
InfLLM	6k	15.0	24.1	41.7	**24.9**	26.5
SnapKV	6k	16.2	25.3	44.0	24.7	27.5
RetroLM-Stage1	6k	18.4	27.8	45.0	24.2	28.9
RetroLM-Stage2	6k	**20.2**	**29.2**	**46.1**	24.5	**30.0**

Table 3: Experiment results on InfiniteBench. The results demonstrate the effectiveness and generalization of RetroLM across ultra-long contexts compared with other efficient processing methods.

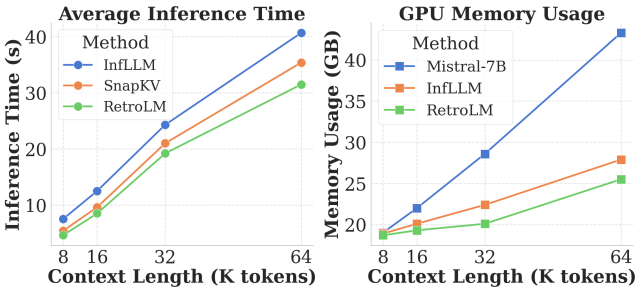

Figure 2: End-to-end generation efficiency analysis.

performs all RAG methods. These findings highlight the superior ability of RetroLM to effectively utilize long-context information, which can be attributed to its dynamic KV retrieval mechanism. Traditional RAG relies on partitioning raw text into arbitrary chunks (e.g., 200 tokens). This process inevitably shatters the document's semantic and structural coherence, often separating related pieces of evidence required for complex reasoning. A retriever's failure to fetch all necessary chunks results in an irrecoverable loss of information (Xu et al. 2023). In contrast, RetroLM operates on the LLM's KV cache, where bookmark tokens represent contiguous but semantically rich pages. This KV-level operation preserves contextual integrity, enabling the model to effectively retrieve and reason over scattered evidence, a crucial advantage demonstrated by its strong performance on multi-hop QA tasks.

Generalization to Ultra-long contexts

We compare RetroLM with other efficient processing methods to demonstrate its effectiveness and generalization across ultra-long contexts. The experimental results on InfiniteBench (Zhang et al. 2024) are presented in Table 3. Given that the lengths of most evaluation cases exceed 100K, we allocated 6K KV budgets for all baselines.

Across all tasks, RetroLM consistently outperforms the full-attention baseline. This indicates that RetroLM effectively generalizes in scenarios involving ultra-long texts, despite being trained on significantly shorter context lengths. Specifically, during Stage-1, the KV retriever was trained on contexts up to 8K tokens, while in Stage-2, the language model was trained with an unsupervised corpus, using a maximum context length of 12K tokens.

When compared to other efficient processing methods, RetroLM demonstrates a clear performance advantage. In the lengthy QA and summarization tasks, RetroLM-Stage2 outperforms SnapKV by 4.0 and 3.9 points respectively. This underscores RetroLM's potential as a scalable and effective solution for real-world applications that require processing of extremely long text.

Models	MMLU	GSM8K
Mistral-7B-Instruct	62.5	35.4
RetroLM-Stage1-Mistral	62.5	35.4
RetroLM-Stage2-Mistral	63.2	34.6

Table 4: Performance comparison on MMLU and GSM8K. The results are reported in terms of accuracy.

Efficiency Analysis

As shown in Figure 2, RetroLM achieves substantially lower memory usage and faster decoding compared to baselines as context length increases. While full attention methods such as Mistral-7B exhibit rapid growth in memory consumption, RetroLM maintains a near-linear scaling by enforcing a fixed KV cache budget via streaming encoding and page retrieval. This design not only reduces peak memory usage but also improves efficiency by implicitly compressing long contexts during inference.

Preservation of General Capabilities

To investigate the impact of RetroLM training on the LLM's foundational abilities, we conduct evaluations on the MMLU (Hendrycks et al. 2020) and GSM8K (Cobbe et al. 2021) benchmarks. As shown in Table 4, we train a specialized, plug-and-play KV retriever while keeping the parameters of the LLM frozen in stage-1. As a result, this stage does not impact the LLM's existing capabilities in any way. The second stage is conducted in an unsupervised manner on a general-domain corpus. This approach aligns with widely adopted practices in prior work (Fu et al. 2024; Gao et al. 2024) on long-context adaptation for LLMs and minimally impacts the model's overall abilities.

Model	Context	Narrative	Qasper	Multifield	Hotpot	2wikim	Musique	Average
Mistral-7B-v0.2	32k	26.9	33.1	49.2	43.0	27.3	18.8	33.1
Ablation Study								
RetroLM w/o Stage1	2k	23.6	29.9	45.4	38.5	24.9	15.1	29.6
RetroLM-Stage1	2k	26.8	34.0	50.8	47.6	39.0	22.5	36.8
RetroLM-Stage2	2k	26.6	38.7	53.8	47.7	41.6	26.4	39.1
Mistral-Finetuned	32k	26.9	33.4	48.5	44.5	30.6	19.4	33.9
InfLLM-Finetuned	2k	25.4	30.7	48.0	43.7	29.2	18.0	32.5
Analytical Experiment with Varying Budgets								
SnapKV (1024)	1024	25.4	29.5	49.0	40.9	25.7	18.3	31.5
SnapKV (2048)	2048	25.9	32.9	48.6	43.0	27.4	19.0	32.8
RetroLM-Stage1 (512)	512	25.0	30.4	47.0	42.9	30.4	17.9	32.3
RetroLM-Stage1 (1024)	1024	25.4	31.5	47.9	45.4	33.7	21.1	34.2
RetroLM-Stage1 (2048)	2048	26.8	34.0	50.8	47.6	39.0	22.5	36.8

Table 5: Analytical experiments with QA tasks from LongBench.

Model	4K	8K	16K	32K	64K	AVG
NIAH Performance						
Mistral-7B-v0.2	98.1	96.2	94.3	85.5	51.1	85.4
RetroLM-Stage2	99.1	96.4	92.2	88.6	79.0	91.1

Table 6: Experiment results of NIAH tasks on RULER.

Information Seeking under Lengthy Context

Beyond downstream long-context understanding tasks such as QA and summarization, we assess long-context information seeking capability of RetroLM using eight Needle-in-a-Haystack tasks from RULER (Hsieh et al. 2024). These tasks cover a diverse range of needle types and quantities with varying levels of difficulty, requiring the model to extract relevant information from a vast number of distractors. As shown in Table 6, RetroLM achieves superior performance compared to the full-attention Mistral model across evaluation lengths ranging from 4K to 64K, demonstrating robust long-context information identification capability.

Case Study

To further evaluate the effectiveness of KV cache level retrieval augmentation in RetroLM, we conduct case study using the MusiQue dataset (Bai et al. 2023), a challenging multi-hop QA task involving lengthy texts. We compare the full attention scores with those of the page retriever. Full attention fails to attend to the KVs containing the correct answer, resulting in an incorrect prediction. In contrast, our proposed page retriever effectively identifies and retrieves the relevant pages. Especially in the intermediate layers, page retriever demonstrates strong ability to focus on crucial KVs. Due to space constraint, more cases are presented in *Appendix Additional Experiments*.

Ablation Study

Effectiveness of Page Retriever. As presented in Table 5 (Top), to assess the effectiveness of page retriever training (Stage1), we implement the algorithmic framework of RetroLM without training the page retriever (w/o Stage1). The resulting test performance exhibits a 6.9 points degradation, underscoring the critical importance of training the page retriever for KV cache level retrieval augmentation.

Effectiveness of Post Training. As presented in Table 5 (Top), to assess the effectiveness of post-training (Stage2), we use the same unsupervised data to perform full-attention fine-tuning and evaluating on the Mistral model (Mistral-Finetuned). We then apply InfLLM (Xiao et al. 2024) algorithm using this model (InfLLM-Finetuned). While these approaches yielded modest performance improvements, they were markedly inferior to the results achieved by RetroLM.

Varying KV Budgets. We assess the effectiveness of RetroLM under varying KV budgets. Using the RetroLM-stage1 model, which only trains the retriever module, we vary the token budget from 512 to 2048 for evaluation. The results are reported in Table 5 (Bottom). Even with 512-token budget, RetroLM achieves an average performance closely aligns with full attention method.

Conclusion

In this paper, we introduce **RetroLM**, a novel framework that enhances the performance of long-context processing by conducting retrieval augmentation at the KV cache level. Unlike traditional RAG methods that operate on raw tokens, RetroLM partitions the KV cache into contiguous pages and selectively retrieves the most crucial ones. To achieve precise retrieval, we propose a specialized **page retriever** that evaluates page importance via fine-grained KV interactions. Additionally, we employ **post-training** on unlabeled data, enabling LLMs to better utilize retrieved KVs and improving end-to-end performance. Extensive evaluations are conducted on several standard long-context benchmarks.

Acknowledgments

This work was supported by Beijing Natural Science Foundation (L243006).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bai, Y.; Lv, X.; Zhang, J.; Lyu, H.; Tang, J.; Huang, Z.; Du, Z.; Liu, X.; Zeng, A.; Hou, L.; et al. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Bajaj, P.; Campos, D.; Craswell, N.; Deng, L.; Gao, J.; Liu, X.; Majumder, R.; McNamara, A.; Mitra, B.; Nguyen, T.; et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Caciularu, A.; Peters, M. E.; Goldberger, J.; Dagan, I.; and Cohan, A. 2023. Peek across: Improving multi-document modeling via cross-document question-answering. *arXiv preprint arXiv:2305.15387*.

Cai, Z.; Zhang, Y.; Gao, B.; Liu, Y.; Liu, T.; Lu, K.; Xiong, W.; Dong, Y.; Chang, B.; Hu, J.; et al. 2024. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*.

Chen, J.; Xiao, S.; Zhang, P.; Luo, K.; Lian, D.; and Liu, Z. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Chen, S.; Wong, S.; Chen, L.; and Tian, Y. 2023a. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.

Chen, Y.; Qian, S.; Tang, H.; Lai, X.; Liu, Z.; Han, S.; and Jia, J. 2023b. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Comanici, G.; Bieber, E.; Schaekermann, M.; Pasupat, I.; Sachdeva, N.; Dhillon, I.; Blistein, M.; Ram, O.; Zhang, D.; Rosen, E.; et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Ding, Y.; Zhang, L. L.; Zhang, C.; Xu, Y.; Shang, N.; Xu, J.; Yang, F.; and Yang, M. 2024. Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Fu, Y.; Panda, R.; Niu, X.; Yue, X.; Hajishirzi, H.; Kim, Y.; and Peng, H. 2024. Data engineering for scaling language models to 128k context. *arXiv preprint arXiv:2402.10171*.

Gao, T.; Wettig, A.; Yen, H.; and Chen, D. 2024. How to train long-context language models (effectively). *arXiv preprint arXiv:2410.02660*.

Günther, M.; Ong, J.; Mohr, I.; Abdessalem, A.; Abel, T.; Akram, M. K.; Guzman, S.; Mastrapas, G.; Sturua, S.; Wang, B.; et al. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *arXiv preprint arXiv:2310.19923*.

Gur, I.; Furuta, H.; Huang, A.; Safdari, M.; Matsuo, Y.; Eck, D.; and Faust, A. 2023. A real-world webagent with planning, long context understanding, and program synthesis. *arXiv preprint arXiv:2307.12856*.

Han, C.; Wang, Q.; Xiong, W.; Chen, Y.; Ji, H.; and Wang, S. 2023. Lm-infinite: Simple on-the-fly length generalization for large language models. *arXiv preprint arXiv:2308.16137*.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Hsieh, C.-P.; Sun, S.; Kriman, S.; Acharya, S.; Rekesh, D.; Jia, F.; Zhang, Y.; and Ginsburg, B. 2024. RULER: What's the Real Context Size of Your Long-Context Language Models? *arXiv preprint arXiv:2404.06654*.

Izacard, G.; Caron, M.; Hosseini, L.; Riedel, S.; Bojanowski, P.; Joulin, A.; and Grave, E. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

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

Jiang, H.; Li, Y.; Zhang, C.; Wu, Q.; Luo, X.; Ahn, S.; Han, Z.; Abdi, A. H.; Li, D.; Lin, C.-Y.; et al. 2024. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. *arXiv preprint arXiv:2407.02490*.

Jiang, Z.; Ma, X.; and Chen, W. 2024. Longrag: Enhancing retrieval-augmented generation with long-context llms. *arXiv preprint arXiv:2406.15319*.

Jin, B.; Yoon, J.; Han, J.; and Arik, S. O. 2024. Long-Context LLMs Meet RAG: Overcoming Challenges for Long Inputs in RAG. *arXiv preprint arXiv:2410.05983*.

Karpukhin, V.; Oğuz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Li, D.; Shao, R.; Xie, A.; Sheng, Y.; Zheng, L.; Gonzalez, J.; Stoica, I.; Ma, X.; and Zhang, H. 2023. How Long Can Context Length of Open-Source LLMs truly Promise? In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Li, X.; Cao, Y.; Ma, Y.; and Sun, A. 2024a. Long Context vs. RAG for LLMs: An Evaluation and Revisits. *arXiv preprint arXiv:2501.01880*.

Li, Y.; Huang, Y.; Yang, B.; Venkitesh, B.; Locatelli, A.; Ye, H.; Cai, T.; Lewis, P.; and Chen, D. 2024b. Snapkv: Llm knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*.

Liu, D.; Chen, M.; Lu, B.; Jiang, H.; Han, Z.; Zhang, Q.; Chen, Q.; Zhang, C.; Ding, B.; Zhang, K.; et al. 2024. Retrievalattention: Accelerating long-context llm inference via vector retrieval. *arXiv preprint arXiv:2409.10516*.

Lu, E.; Jiang, Z.; Liu, J.; Du, Y.; Jiang, T.; Hong, C.; Liu, S.; He, W.; Yuan, E.; Wang, Y.; et al. 2025. Moba: Mixture of block attention for long-context llms. *arXiv preprint arXiv:2502.13189*.

Luo, K.; Liu, Z.; Xiao, S.; and Liu, K. 2024. BGE Landmark Embedding: A Chunking-Free Embedding Method For Retrieval Augmented Long-Context Large Language Models. *arXiv preprint arXiv:2402.11573*.

Mohtashami, A.; and Jaggi, M. 2023. Landmark attention: Random-access infinite context length for transformers. *arXiv preprint arXiv:2305.16300*.

OpenAI. 2024. Learning to Reason with LLMs. https://openai.com/index/learning-to-reason-with-llms/. Accessed: 2024-12-18.

Peng, B.; Quesnelle, J.; Fan, H.; and Shippole, E. 2023. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*.

Qian, H.; Liu, Z.; Mao, K.; Zhou, Y.; and Dou, Z. 2024. Grounding Language Model with Chunking-Free In-Context Retrieval. *arXiv preprint arXiv:2402.09760*.

Robertson, S.; Zaragoza, H.; et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4): 333–389.

Shen, Z.; Tao, T.; Ma, L.; Neiswanger, W.; Liu, Z.; Wang, H.; Tan, B.; Hestness, J.; Vassilieva, N.; Soboleva, D.; et al. 2023. Slimpajama-dc: Understanding data combinations for llm training. *arXiv preprint arXiv:2309.10818*.

Tang, J.; Zhao, Y.; Zhu, K.; Xiao, G.; Kasikci, B.; and Han, S. 2024. Quest: Query-Aware Sparsity for Efficient Long-Context LLM Inference. *arXiv preprint arXiv:2406.10774*.

Xiao, C.; Zhang, P.; Han, X.; Xiao, G.; Lin, Y.; Zhang, Z.; Liu, Z.; Han, S.; and Sun, M. 2024. Infllm: Unveiling the intrinsic capacity of llms for understanding extremely long sequences with training-free memory. *arXiv preprint arXiv:2402.04617*.

Xiao, G.; Tian, Y.; Chen, B.; Han, S.; and Lewis, M. 2023a. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Xiao, S.; Liu, Z.; Zhang, P.; and Muennighof, N. 2023b. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*.

Xu, P.; Ping, W.; Wu, X.; McAfee, L.; Zhu, C.; Liu, Z.; Subramanian, S.; Bakhturina, E.; Shoeybi, M.; and Catanzaro, B. 2023. Retrieval meets long context large language models. *arXiv preprint arXiv:2310.03025*.

Xu, Y.; Jie, Z.; Dong, H.; Wang, L.; Lu, X.; Zhou, A.; Saha, A.; Xiong, C.; and Sahoo, D. 2024. Think: Thinner key cache by query-driven pruning. *arXiv preprint arXiv:2407.21018*.

Yuan, J.; Gao, H.; Dai, D.; Luo, J.; Zhao, L.; Zhang, Z.; Xie, Z.; Wei, Y.; Wang, L.; Xiao, Z.; et al. 2025. Native sparse attention: Hardware-aligned and natively trainable sparse attention. *arXiv preprint arXiv:2502.11089*.

Yue, Z.; Zhuang, H.; Bai, A.; Hui, K.; Jagerman, R.; Zeng, H.; Qin, Z.; Wang, D.; Wang, X.; and Bendersky, M. 2024. Inference scaling for long-context retrieval augmented generation. *arXiv preprint arXiv:2410.04343*.

Zhang, P.; Xiao, S.; Liu, Z.; Dou, Z.; and Nie, J.-Y. 2023a. Retrieve anything to augment large language models. *arXiv preprint arXiv:2310.07554*.

Zhang, X.; Chen, Y.; Hu, S.; Xu, Z.; Chen, J.; Hao, M.; Han, X.; Thai, Z.; Wang, S.; Liu, Z.; et al. 2024. InfBench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 15262–15277.

Zhang, Z.; Sheng, Y.; Zhou, T.; Chen, T.; Zheng, L.; Cai, R.; Song, Z.; Tian, Y.; Ré, C.; Barrett, C.; et al. 2023b. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36: 34661–34710.

Zhao, P.; Zhang, H.; Yu, Q.; Wang, Z.; Geng, Y.; Fu, F.; Yang, L.; Zhang, W.; and Cui, B. 2024. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*.