Does RAG Really Perform Bad In Long-Context Processing?

Anonymous ACL submission

Abstract

The efficient processing of long context poses a serious challenge for large language models (LLMs). Recently, retrieval-augmented generation (RAG) has emerged as a promising strategy for this problem, as it enables LLMs to make selective use of the long context for efficient computation. However, existing RAG approaches lag behind other long-context processing methods due to inherent limitations on inaccurate retrieval and fragmented contexts. To address these challenges, we introduce **RetroLM**, a novel RAG framework for long-context processing. Unlike traditional methods, *RetroLM employs KV-level retrieval augmentation*, where it partitions the LLM's KV cache into contiguous pages and retrieves the most crucial ones for efficient computation. This approach enhances robustness to retrieval inaccuracy, facilitates effective utilization of fragmented contexts, and saves the cost from repeated computation. Building on 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. We conduct comprehensive evaluations with a variety of benchmarks, including LongBench, InfiniteBench, and RULER, where RetroLM significantly outperforms existing long-context LLMs and efficient long-context processing methods, particularly in tasks requiring intensive reasoning or extremely long-context comprehension.

1 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 Gemini-1.5-pro (Team et al., 2024) makes a dramatic extension, enabling a context window of over 10M input tokens.

Despite these advancements, the naive extension of context lengths remains constrained in several aspects. One significant challenge is the dramatic rise in computation when processing long contexts. As such, efficient long-context processing techniques have attracted growing interest. For example, StreamingLLM and LM-Infinite (Xiao et al., 2023a; Han et al., 2023) maintain the most recent KVs within a sliding window alongside initial attention sinks; while SnapKV and InfLLM (Li et al., 2024b; Xiao et al., 2024) identify critical attention features for KV compression. Recently, retrieval-augmented generation (RAG) has emerged as a promising strategy for this problem (Xu et al., 2023; Li et al., 2024a). These approaches leverage retrievers to extract useful context fragments from very long inputs, which effectively overcomes the limits of LLMs' context lengths. By making selective use of the retrieved fragments, RAG further enables more efficient computation for long-context tasks.

However, RAG-based methods are subject to the following inherent limitations while handling long-context tasks. 1) *Retrieval Inaccuracy*. Many long-context processing tasks provide no explicit queries at all, like document summarization, code completion, and in-context learning (Bai et al., 2023). As a result, traditional retrievers become inapplicable to handle corresponding problems. Besides, it's non-trivial to properly chunk long contexts for retrieval (Qian et al., 2024), and it's hard for retrievers to

deal with zero-shot settings. Without precise and complete acquisition of useful information, LLMs will be unable to produce correct outputs through RAG. 2) *Fragmented Contexts*. The retrieval operation introduces fragmented token spans from the input data, which are incoherent and prone to incompleteness. This significantly prevents LLMs from making effective use of the contextual information. 3) *Repeated Computation*. The pre-filling operation needs to be re-conducted for the retrieved tokens of each task, both for retriever and generator, resulting in a huge waste of computation. *Because of the above problems, existing RAG-based methods fall behind long-context LLMs and other efficient long-context processing approaches in many popular evaluation benchmarks* (Bai et al., 2023; Xu et al., 2023; Li et al., 2024a).

In this paper, we propose **RetroLM**, a novel RAG framework designed for efficient long-context processing. RetroLM partitions the LLM's KV cache into contiguous pages and offloads them to external storage. During both pre-filling and decoding stages, it retrieves only the most crucial pages for the current context window, enabling efficient long-context processing. Unlike traditional RAG approaches which operate on raw tokens, retrieval augmentation at the KV cache level offers several advantages. First, it is robust to retrieval inaccuracy, as useful information within a certain token span can be captured by all succeeding KVs. Second, LLMs can naturally accommodate fragmented KVs due to the inherent sparsity of LLMs' attention patterns (Jiang et al., 2024). Third, the KV cache is computed once and reused, thus eliminating repeated computation (Pope et al., 2023).

We introduce a couple of key operations to optimize the performance of RetroLM. For precise retrieval of crucial pages, we design a specialized **page retriever**. It estimates the pages' importance using fine-grained KV interactions; and by fine-tuning over well-curated datasets, it achieves strong generality across various downstream tasks and a broad scope of context lengths. To make better use of fragmented KVs, we perform **post-training** based on unlabeled data. This further contributes to the end-to-end performance of RetroLM.

We perform comprehensive evaluations using several standard benchmarks in this field, including LongBench (Bai et al., 2023), InfiniteBench (Zhang et al., 2024b), 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 noise and focusing on the most useful KV entries. Our well-trained models and source code will be made publicly available to facilitate future research.

2 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 approaches 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; Xiong et al., 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).

Recent studies have explored various types of efficient long-context processing techniques to alleviate computational and memory constraints (Sun et al., 2024; Liao et al., 2024; Yang et al., 2024). Stream processing approaches, such as StreamingLLM (Xiao et al., 2023a) and LM-Infinite (Han et al., 2023), maintain the most recent KVs within a sliding window alongside initial attention sinks. Sequential compression techniques, such as Activation Beacon (Zhang et al., 2024a), compress intermediate activations into more compact forms to conserve memory. KV quantization methods, including KIVI (Liu et al., 2024b), encode the KV cache using low-bit representations to minimize storage requirements. Among these methods, KV cache sparsification has gained significant attention for their ability to selectively uti-

lize 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; Huang et al., 2024; Liu et al., 2024a; Shi et al., 2024). 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. H2O (Zhang et al., 2023) introduces a policy that greedily drops KVs during generation using a scoring function derived from cumulative attention. 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.

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.

3 Method

3.1 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 target output 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, 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 $p^{ret}(\cdot)$. 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 methods to approximate full attention, RetroLM introduces a specialized and trainable page retriever, inspired by dense retrieval techniques. We conduct further analysis in Sec. 4.7 to demonstrate the effectiveness of page retriever over full attention.

3.2 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). Then for each page X_i, a special bookmark token ($\langle$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:

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

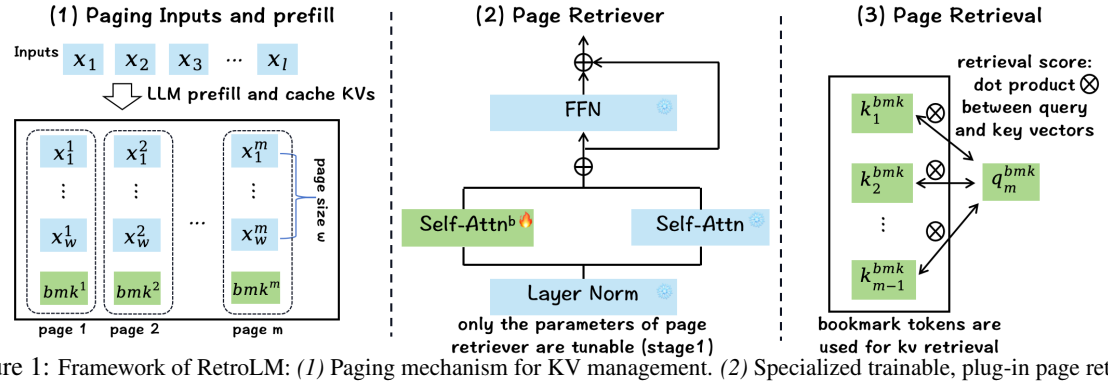

Figure 1: Framework of RetroLM: *(1)* Paging mechanism for KV management. *(2)* Specialized trainable, plug-in page retriever. *(3)* KV retrieval using special bookmark tokens, with their representation established within attention module.

3.3 Page Retriever

Architecture. We 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:

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

where $\boldsymbol{W}_*^n$ are the LLM's original projection matrices and $\boldsymbol{W}_*^b$ 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:

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

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

Training. Training the page retriever poses a challenge due to the lack of appropriately labeled long-context data for retrieval supervision signals. Drawn inspiration from the training paradigm of advanced dense retrievers, where a few negative samples are employed to establish the ability to distinguish relevant passages from corpus containing millions of samples, we adopt contrastive learning to train the page retriever for strong generalizability and robustness (Karpukhin et al., 2020; Chen et al., 2024; Luo et al., 2024).

As shown in Figure 5, we leverage 50K pairwise data from the dense retrieval training set MS MARCO (Bajaj et al., 2016), which is derived from real-world web search queries, to provide valuable transferable semantic matching capabilities for the page retriever. To construct input sequences, we concatenate the positive passage with hard negative passages in a random order, forming pseudo-texts up to a length of 8K tokens, and append the web search query to the end of the sequence (page X_m'). Additionally, we synthesize 5K pairwise samples using text from Slimpajama (Shen et al., 2023), which contains coherent contexts that enable page retriever to effectively learn to find target KVs relevant to query. The detailed data format and training implementation are described in Appendix A.

Assuming the useful KVs for the local page m (where the query resides) is located on page i (denoted as X_i') the contrastive learning objective is defined as follows:

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

where $\boldsymbol{q}_*^{bmk}$ and $\boldsymbol{k}_*^{bmk}$ are the query and key vectors of bookmark tokens of corresponding pages in the self-attention module. This training phase, referred to as **Stage-1**, focuses on training the page retriever to identify useful KVs against complex and distracting contexts at each decoder layer, while keeping the backbone LLM frozen.

3.4 Post Training

We conduct **Stage-2** post-training for RetroLM, during which model parameters are fine-tuned to adapt to sparse KV caches retrieved by the page retriever. Our training leverages unsupervised pre-training data from SlimPajama (Shen et al., 2023)

(up to 12K tokens) and employs a streaming encoding strategy. During language modeling, each page uses the well-trained page retriever to select most semantically important top-k pages for attention computation (similar to inference process), instead of relying on full attention:

$$C : \{X'_1, ..., X'_k\} = p^{kv}(\{X'_1, ..., X'_{i-1}\}|X'_i) \quad (8)$$

The loss function follows the standard language modeling loss formula:

$$L_2 = -\sum_t \log P(x_t|x_{<t}) \quad (9)$$

The training data consists of unsupervised, relatively short-length corpus segments from SlimPajama (Shen et al., 2023), with a maximum token length of 12K. Notably, the objective is not length extension but rather enhancing the model's capacity for adaptation to retrieved sparse KV pages. We use the same data to finetune LLM directly for further analysis in Sec. 4.8.

4 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. 2) How well can RetroLM generalize to different long-context tasks and context lengths.

4.1 Setting

Datasets. To comprehensively evaluate the overall performance of RetroLM, we employ the LongBench suite (Bai et al., 2023). This benchmark encompasses a variety of tasks, including single-document QA, multi-hop QA, summarization, and long ICL. These tasks are well-suited for assessing the long-context capability of different methods in practical application scenarios. Subsequently, to assess the generalization of RetroLM in extremely long scenarios, we utilize several realistic and representative tasks from InfiniteBench (Zhang et al., 2024b), 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 (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 Sparsification: These methods, such as H2O (Zhang et al., 2023), 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 et al., 2009), the Contriever model (Izacard et al., 2021), and the SOTA BGE-large-v1.5 model (Xiao et al., 2023b).

4.2 Comparing with other Efficient Processing Methods on LongBench

We compare stage-1 and stage-2 training of RetroLM with other efficient processing methods, using two popular backbone LLM (**Mistral-7B-Instruct** and **Llama-3-8B-Instruct**). The results on LongBench (Bai et al., 2023) 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

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. The result emphasizes the effectiveness of RetroLM over strong baselines and a wide variety of tasks.

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 of comparing RetroLM with RAG methods on LongBench QA tasks.

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 (see Sec. 4.8). Similar trends are observed in experiments with the Llama-3-based models, corroborating the generality of our findings.

4.3 Comparing with RAG on LongBench

In this section, we compare RetroLM with retrieval-augmented generation (RAG) methods, 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 similar chunks (each 200 tokens) for each dataset.

The results demonstrate that RetroLM consistently outperforms 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. Unlike RAG methods, which rely on a static selection of information at the input stage, RetroLM dynamically retrieves crucial KVs at each decoder layer. This dynamic approach enables RetroLM to preserve crucial information and maintain global contextual awareness. While RAG's rigid prompt selection often leads to the permanent loss of relevant information that the retriever fails to identify (Xu et al., 2023). Moreover, RAG's method for handling long-context tasks necessitates additional retrieval models and stages, thereby increasing the complexity of the task flow. In contrast, using RetroLM for long-context tasks allows for an end-to-end approach using a single model.

RetroLM's training of KV retriever in Stage-1 draws inspiration from contrastive learning paradigms employed in general-purpose dense retriever training (Izacard et al., 2021; Ma et al., 2024). This design allows it to accurately identify useful KV pages from complex, noisy contexts (hard negatives). As a result, RetroLM can be viewed as a novel model-based RAG framework, integrating retrieval functionality directly into the LLM at KV cache level. This integration enhances both semantic comprehension and portability. We believe this approach holds broader research value in the future, including extensions to more complex long-context reasoning scenarios and the development of knowledge bases capable of storing and retrieving KVs.

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.

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
GPU Memory (GB)						
Mistral-7B-v0.2	17.0G	19.0G	22.0G	28.6G	43.3G	-
RetroLM-Stage2	18.3G	18.7G	19.3G	20.1G	25.5G	-

Table 4: Experiment results of NIAH tasks on RULER and GPU memory usage at different input lengths.

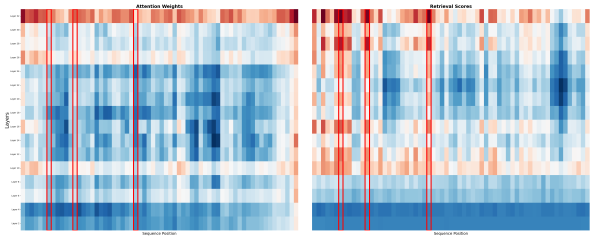

Figure 2: Attention score maps for a MusiQue case. Left: from original full attention. Right: score from RetroLM's KV retriever. Red squares are answers for the multi-hop question. X-axis represents sequence position, Y-axis represents each decoder layer. RetroLM effectively retrieves crucial KVs.

4.4 Experiment Results on InfiniteBench

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

4.5 Experiment Results on RULER

Beyond downstream long-context understanding tasks such as QA and summarization, we assess long-context retrieval 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 4 (Top), 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.

4.6 GPU Memory Consumption

As shown in Table 4 (Bottom), RetroLM significantly reduces memory consumption compared to full attention as input length increases. The memory usage of full attention grows quadratically with sequence length, while even flash attention exhibits linear growth. In contrast, RetroLM leverages streaming encoding with the page retriever to maintain a fixed KV budget during both the prefilling and decoding. This approach effectively minimizes peak memory consumption and implicitly compresses the context.

4.7 Case Study

To further evaluate the effectiveness of KV cache level retrieval augmentation in RetroLM, we conduct case study using the MusiQue dataset (Bai

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.

et al., 2023), a challenging multi-hop QA task involving lengthy texts. As illustrated in Figure 2, 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 B.

4.8 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 after Stage-2 training. This demonstrates the necessity of adapting the model to sparse KV cache for effective KV cache usage and enhanced performance.

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 vray the token budget from 512 to 2048 and evaluate on the Longbench QA datasets. For comparative analysis, we include results from SnapKV with 1024 and 2048 token budgets and the full-attention model. The results are reported in Table 5 (Bottom).

Even with 512-token budget, RetroLM achieves an average score of 32.3 across the LongBench QA tasks, closely aligns with both SnapKV using a 2048-token budget and the full-attention model. As the token budget increases, we observe a clear trend of performance improvement. The significant performance gains on complex datasets like 2WikiMQA (+8.6) and HotpotQA (+6.7) suggest the effectiveness of RetroLM in complex long-context reasoning scenarios that demand robust information seeking and aggregation capabilities.

5 Conclusion

In this paper, we introduce **RetroLM**, a novel RAG 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 and effective 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.

6 Limitation

While RetroLM achieves substantial progress in efficient long-context processing, computational constraints and design choices have led to the use of a relatively small base model (7B) during the experimental phase. It is anticipated that scaling to larger models could further enhance performance. Additionally, the fixed page size also presents a limitation; exploring dynamic or context-dependent page sizes could optimize the trade-off between granularity and computational cost.

7 Ethical consideration

RetroLM is built upon open-source LLMs. Consequently, it inherits similar ethical and social risks, such as bias, discrimination, and the potential for generating toxic or harmful content, as those associated with the base LLM. The pre-training data of the base LLM may contain private or sensitive information, posing a low but non-zero risk of information leakage, despite RetroLM operating on the KV cache. Furthermore, the page retriever's fine-tuning data could introduce biases in the retrieval process, favoring certain types of information.

References

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

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Avi Caciularu, Matthew E Peters, Jacob Goldberger, Ido Dagan, and Arman Cohan. 2023. Peek across: Improving multi-document modeling via cross-document question-answering. *arXiv preprint arXiv:2305.15387*.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Baobao Chang, Junjie Hu, et al. 2024. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*.

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 preprint arXiv:2402.03216*.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023a. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023b. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*.

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 preprint arXiv:2407.21783*.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. 2024. Data engineering for scaling language models to 128k context. *arXiv preprint arXiv:2402.10171*.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2024. How to train long-context language models (effectively). *arXiv preprint arXiv:2410.02660*.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, et al. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *arXiv preprint arXiv:2310.19923*.

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. 2023. A real-world webagent with planning, long context understanding, and program synthesis. *arXiv preprint arXiv:2307.12856*.

Chi Han, Qifan Wang, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2023. Lm-infinite: Simple on-the-fly length generalization for large language models. *arXiv preprint arXiv:2308.16137*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Yuxiang Huang, Binhang Yuan, Xu Han, Chaojun Xiao, and Zhiyuan Liu. 2024. Locret: Enhancing eviction in long-context llm inference with trained retaining heads. *arXiv preprint arXiv:2410.01805*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H Abdi, Dongsheng Li, Chin-Yew Lin, et al. 2024. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. *arXiv preprint arXiv:2407.02490*.

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O Arik. 2024. Long-context llms meet rag: Overcoming challenges for long inputs in rag. *arXiv preprint arXiv:2410.05983*.

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 preprint arXiv:2004.04906*.

Dacheng Li, Rulin Shao, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. 2023. How long can context length of open-source llms truly promise? In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Xinze Li, Yixin Cao, Yubo Ma, and Aixin Sun. 2024a. Long context vs. rag for llms: An evaluation and revisits. *arXiv preprint arXiv:2501.01880*.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024b. Snapkv: Llm knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*.

Zihan Liao, Jun Wang, Hang Yu, Lingxiao Wei, Jianguo Li, and Wei Zhang. 2024. E2llm: Encoder elongated large language models for long-context understanding and reasoning. *arXiv preprint arXiv:2409.06679*.

Di Liu, Meng Chen, Baotong Lu, Huiqiang Jiang, Zhenhua Han, Qianxi Zhang, Qi Chen, Chengruidong Zhang, Bailu Ding, Kai Zhang, et al. 2024a. Retrievalattention: Accelerating long-context llm inference via vector retrieval. *arXiv preprint arXiv:2409.10516*.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024b. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*.

Kun Luo, Zheng Liu, Shitao Xiao, and Kang Liu. 2024. Bge landmark embedding: A chunking-free embedding method for retrieval augmented long-context large language models. *arXiv preprint arXiv:2402.11573*.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2421–2425.

Amirkeivan Mohtashami and Martin Jaggi. 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.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. 2023. Efficiently scaling transformer inference. *Proceedings of Machine Learning and Systems*, 5:606–624.

Hongjin Qian, Zheng Liu, Kelong Mao, Yujia Zhou, and Zhicheng Dou. 2024. Grounding language model with chunking-free in-context retrieval. *arXiv preprint arXiv:2402.09760*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

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

Zhiqiang Shen, Tianhua Tao, Liqun Ma, Willie Neiswanger, Zhengzhong Liu, Hongyi Wang, Bowen Tan, Joel Hestness, Natalia Vassilieva, Daria Soboleva, et al. 2023. Slimpajama-dc: Understanding data combinations for llm training. *arXiv preprint arXiv:2309.10818*.

Zhenmei Shi, Yifei Ming, Xuan-Phi Nguyen, Yingyu Liang, and Shafiq Joty. 2024. Discovering the gems in early layers: Accelerating long-context llms with 1000x input token reduction. *arXiv preprint arXiv:2409.17422*.

Hanshi Sun, Li-Wen Chang, Wenlei Bao, Size Zheng, Ningxin Zheng, Xin Liu, Harry Dong, Yuejie Chi, and Beidi Chen. 2024. Shadowkv: Kv cache in shadows for high-throughput long-context llm inference. *arXiv preprint arXiv:2410.21465*.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. 2024. Quest: Query-aware sparsity for efficient long-context llm inference. *arXiv preprint arXiv:2406.10774*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, Song Han, and Maosong Sun. 2024. Infllm: Unveiling the intrinsic capacity of llms for understanding extremely long sequences with training-free memory. *arXiv preprint arXiv:2402.04617*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023a. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023b. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. 2023. Effective long-context scaling of foundation models. *arXiv preprint arXiv:2309.16039*.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Retrieval meets long context large language models. *arXiv preprint arXiv:2310.03025*.

Yuhui Xu, Zhanming Jie, Hanze Dong, Lei Wang, Xudong Lu, Aojun Zhou, Amrita Saha, Caiming Xiong, and Doyen Sahoo. 2024. Think: Thinner key cache by query-driven pruning. *arXiv preprint arXiv:2407.21018*.

Dongjie Yang, XiaoDong Han, Yan Gao, Yao Hu, Shilin Zhang, and Hai Zhao. 2024. Pyramidinfer: Pyramid kv cache compression for high-throughput llm inference. *arXiv preprint arXiv:2405.12532*.

Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. 2024. Inference scaling for long-context retrieval augmented generation. *arXiv preprint arXiv:2410.04343*.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. 2024a. Soaring from 4k to 400k: Extending llm's context with activation beacon. *arXiv preprint arXiv:2401.03462*, 2(3):5.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, et al. 2024b. 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)*, pages 15262–15277.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.

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 preprint arXiv:2402.19473*.

A Training Detail and Data Formulation

A.1 Data Formulation

In this section, we introduce the detailed data curating method for the training of RetroLM. For stage-1 training, we exclusively train the KV retriever while keeping the backbone LLM fixed. We utilize a dataset of 50K pairwise examples sourced from MS MARCO (Bajaj et al., 2016), formatted as illustrated in Figure 5. Each dataset entry comprises a web search query, along with 40 hard negative passages mined by BGE (Xiao et al., 2023b). These passages are randomly interspersed with a positive passage, resulting in pseudo input texts up to 8,000 tokens in length. The query is appended at the end of the text, prompting the model to locate pertinent information corresponding to the input query. Leveraging the robust semantic understanding of the fixed backbone LLM, we employ contrastive learning across each decoder layer using bookmark tokens. This task requires the KV retriever to discern key KVs amidst substantial distractions. The input constraint of 8K tokens ensures high training efficiency and aims to establish the retrieval proficiency of the KV retriever.

Additionally, we generate 5K synthetic pairwise samples using text from Slimpajama (Shen et al., 2023), structured as depicted in Figure 6. Unlike discrete text spans from MS MARCO, these input texts consist of coherent passages, tasking the KV retriever with identifying key KVs for the query. For the detailed curation method, we begin by sampling lengthy documents from Slimpajama. From these documents, we extract a segment (e.g., 100 words) as the *Background Text*, and randomly select consecutive 1-5 sentences from this segment as the *Ground Truth Text*. Employing the GPT API[1], we pose questions about the *Background Text*, stipulating that the answers must be contained within the *Ground Truth Text*. This method ensures that synthetic questions are contextually rich while ensuring that their answers remain within smaller semantic units. The prompt for constructing synthetic data are provided in Figure 7. To maintain the quality of synthetic data, we ask ChatGPT to generate precise and insightful questions. If the generated text lacks meaningful information, it undergoes careful scrutiny and filtration.

For stage-2 training, we use 10K unsupervised text data from Slimpajama (Shen et al., 2023), with input lengths constrained to 12K tokens. The primary goal of stage-2 training is not to expand the context window but to enable RetroLM to adapt to a sparse KV cache and a streaming encoding paradigm. The corresponding ablation study is detailed in Sec. 4.8.

A.2 Implementation

In this Section, we introduce the training and implementation details of RetroLM. We train RetroLM using **Mistral-7B-Instruct** and **Llama-3-8B-Instruct** as backbone respectively. During training, we set the page size for RetroLM to 128 tokens. This means the input text is divided into segments of 128 tokens, each appended with a bookmark token. It is important to note that the page size used during inference does not need to match the training page size; users can define it at any desired granularity.

All the experiments take place on 8xA800 (80GB) GPUs. The learning rate for stage-1 training is 5×10^{-6} and for stage-2 is 1×10^{-6}, the weight decay is 1×10^{-2}. The batch size is 1, where we accumulate the gradient over 16 steps. We leverage Flash-attention-v2 (Dao, 2023), Gradient Checkpointing (Chen et al., 2016), and Deepspeed-Zero (Rajbhandari et al., 2020) to speed up the training. Throughout training, the peak CUDA memory usage is observed at about 40GB, which is attributable to the limited input length of 12K tokens.

B Further Case Study

In this section, we present more cases (Figure 3 to Figure 4) to evaluate the effectiveness of KV cache level retrieval augmentation in RetroLM, using the MusiQue dataset (Bai et al., 2023).

[1] https://platform.openai.com/

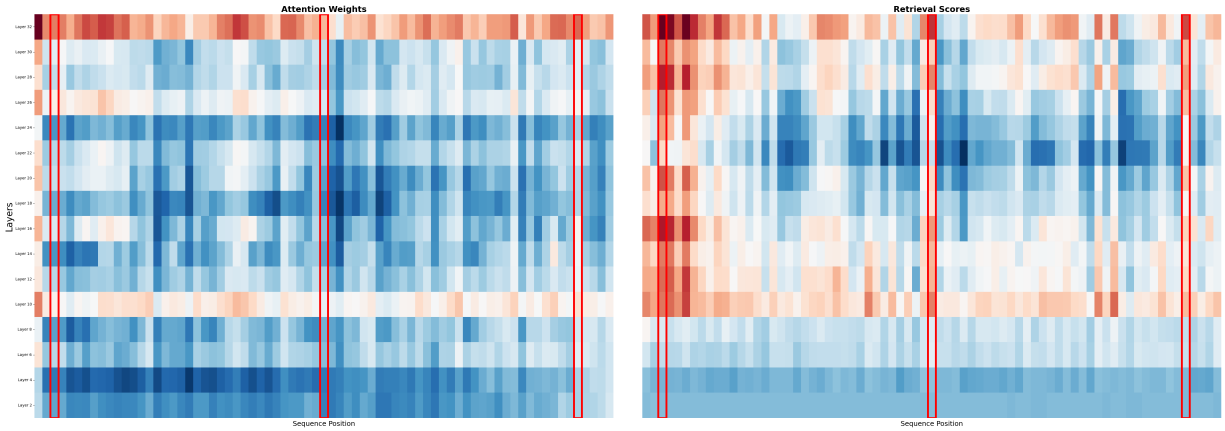

Figure 3: Case 2: attention score maps. Left: from original full attention. Right: score from RetroLM's KV retriever. Red squares are answers for the multi-hop question. X-axis represents sequence position, Y-axis represents each decoder layer. RetroLM effectively retrieves crucial KVs.

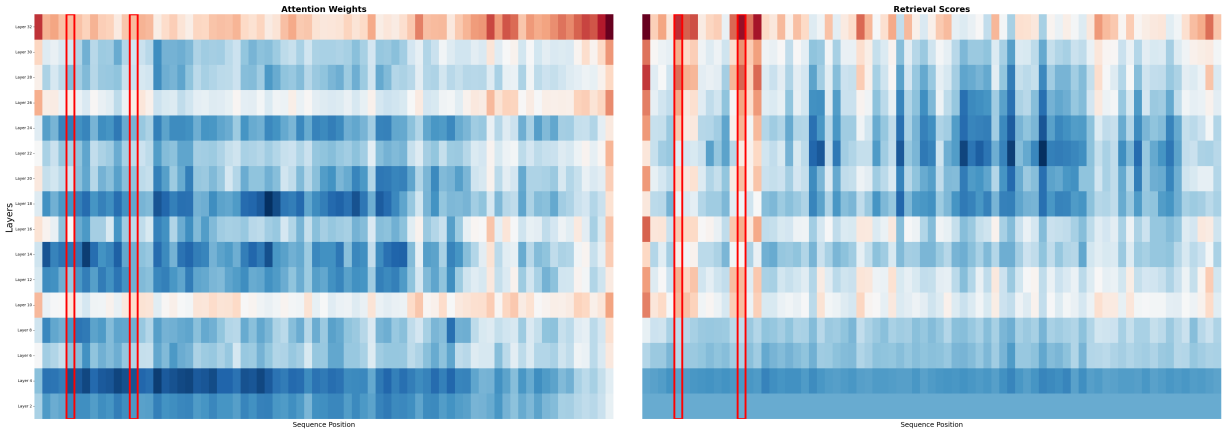

Figure 4: Case 3: attention score maps. Left: from original full attention. Right: score from RetroLM's KV retriever. Red squares are answers for the multi-hop question. X-axis represents sequence position, Y-axis represents each decoder layer. RetroLM effectively retrieves crucial KVs.

Input:

Read the following text and find key information for the question.

(Negative) Owen is portrayed as a roman cavalry officer also known as artorius castus the son of a roman father and a celtic mother who commands a unit of sarmatian auxiliary cavalry ...

...

(Positive) Vaughn in the 2005 film Wedding Crashers, which grossed over US$ 200M in the U.S. alone. Also in 2005, Owen collaborated with his brothers by appearing in The Wendell Baker Story, written by brother Luke, directed by Luke and brother Andrew.

...

(Negative) But the hospitalization this week of Owen Wilson, 38, after police responded to a report of a suicide attempt at his Santa Monica home, astonished anyone who knows him simply as the affable ...

Now, find key information for the question.

Question: who is owen wilson's brothers?

Figure 5: Example of weakly supervised data from msmarco (Bajaj et al., 2016) for stage-1 training. We concatenate the positive passage with hard negative passages in a random order, forming pseudo-texts up to a length of 8K tokens, and append the query to the end of the sequence. **The page retriever is trained to identify useful KVs (positive passage) across each decoder layer via contrastive learning.**

Input:

Read the following text and find key information for the question.

Friday, February 1, 2013 Memo to Washington: Foreign Policy Begins Abroad - Nader Mousavizadeh, New York Times: \"John Kerry's overwhelming confirmation as the next U.S. ...

...

(/Positive) Axiomatic now as the only alternative to doing nothing is the use of lethal force backed by the occasional choice of diplomacy as clean-up job. This is unworthy of a great power — and a great foreign service. During his confirmation hearings, Kerry stated that 'American foreign policy is not defined by drones and deployments alone.' **(Positive/)**

...

Cultural affairs officer soon comes to realize that his job is really a form of love-making and that making love is never really successful unless both partners are participating.

Now, find key information for the question.

Question: What is the author's opinion on the current approach to diplomacy and the use of force?

Figure 6: Example of synthetic data for stage-1 training.

##Please generate a valuable "Question" based on the provided "Background Text" and make sure the "Question" has an answer in the "Ground Truth Text", note that the "Ground Truth Text" is part of "Background Text".

##Principle:

Note that the "Ground Truth Text" is part of the "Background Text", here is more detailed explanation you should understand and follow. The "Ground Truth Text" should contains the answer of the "Question" you generate, which means the answer can be directly answer using the "Ground Truth Text" or can be infered from the "Ground Truth Text". The "Background Text" aims to give you more background information helping you generate more abstractive and valuable "Question".

##Example:

"Background Text": Corruption is rife throughout the economy and the country remains heavily dependent on the oil sector, which in 2017 accounted for over 90 percent of exports by value and 64 percent of government revenue. With the end of oil boom, from 2015 Angola entered into period of economic contraction. The Angolan economy has been dominated by the production of raw materials and the use of cheap labor. The Portuguese used Angola principally as a source for the thriving slave trade across the Atlantic; Luanda became the greatest slaving port in Africa.

"Ground Truth Text": With the end of oil boom, from 2015 Angola entered into period of economic contraction.The Angolan economy has been dominated by the production of raw materials and the use of cheap labor.

"Question": What leads to Angola economic contraction since 2015?

##Now follow the principles generate to the required "Query" with the given "Background Text" and "Ground Truth Text":

"Background Text": {}

"Ground Truth Text": {}

"Question":

Figure 7: Prompt for construct question from Slimpagama (Shen et al., 2023).