*ECAI 2024*
*U. Endriss et al. (Eds.)*


*doi:10.3233/FAIA240965*

# FltLM: An Intergrated Long-Context Large Language Model for Effective Context Filtering and Understanding

**Jingyang Deng[a], Zhengyang Shen[b,*], Boyang Wang[c], Lixin Su[b], Suqi Cheng[b], Ying Nie[b], Junfeng Wang[b], Dawei Yin[b] and Jinwen Ma[a,**]**

[a]School of Mathematical Sciences and LMAM, Peking University
[b]Baidu, Inc.
[c]CCSE, Beihang University

**Abstract.** The development of Long-Context Large Language Models (LLMs) has markedly advanced natural language processing by facilitating the process of textual data across long documents and multiple corpora. However, Long-Context LLMs still face two critical challenges: The *lost in the middle* phenomenon, where crucial middle-context information is likely to be missed, and the *distraction* issue that the models lose focus due to overly extended contexts. To address these challenges, we propose the Context Filtering Language Model (FltLM), a novel integrated Long-Context LLM which enhances the ability of the model on multi-document question-answering (QA) tasks. Specifically, FltLM innovatively incorporates a context filter with a soft mask mechanism, identifying and dynamically excluding irrelevant content to concentrate on pertinent information for better comprehension and reasoning. Our approach not only mitigates these two challenges, but also enables the model to operate conveniently in a single forward pass. Experimental results demonstrate that FltLM significantly outperforms supervised fine-tuning and retrieval-based methods in complex QA scenarios, suggesting a promising solution for more accurate and reliable long-context natural language understanding applications.

## 1 Introduction

The advent of Long-Context Large Language Models (LLMs) marks a significant advancement in natural language processing, addressing the increasing demand for comprehending and generating extensive textual data. The need for such models arises from the vast amount of information contained in lengthy documents and multiple corpora, which general LLMs often struggle to process. Long-context LLMs promise to revolutionize a range of applications, including in-depth cross-document question answering [1], comprehensive document summarization [2] and sophisticated content generation [10, 36], exhibiting a promising future with huge development potential.

Current research has introduced various strategies to extend context window of LLMs, ranging from modifications in positional encoding during the continual pre-training stage, such as Positional Interpolation (PI) [4], NTK-aware interpolation, and YaRN [21], to efficient training methods such as LongLoRA [6], LongQLoRA [31], and PoSE [37]. Furthermore, to address the computational challenges posed by the quadratic complexity of the self-attention mechanism, distributed training approaches have been developed. These techniques, such as sequence parallelism [16] and distributed attention mechanisms, enable scaling up both model size and the context window length, thereby enhancing the processing capacity of LLMs.

With the help of these advancements, the research community has witnessed the emergence and open-sourcing of numerous Long-Context LLMs. However, the evaluation of these models has mainly focused on their long-context modeling capabilities, with perplexity as a metric, or on some basic information retrieval tasks such as pass-key retrieval [4] and the Needle in the Haystack [9]. Relative less attention has been paid to enhancing model performance in downstream real-world tasks, while it is crucial for unlocking the full potential of these models. To get further in this field, we mainly focus on a challenging yet ubiquitous multi-hop multi-document question-answering (QA) task, which asks the model to gather and integrate information from multiple pieces of text to generate a correct answer. Unlike single-hop QA tasks, where the answer to a question can be found straightforwardly within a single sentence or document, multi-hop QA involves reasoning across several documents or parts of a context to synthesize the answer.

Unfortunately, Long-Context LLMs face significant challenges in such tasks, as demonstrated by recent research showing that they struggle with intricate long dependency tasks [14], such as QA tasks that requires model to retrieve multiple pieces of information or engage in comprehension and reasoning. We believe that these challenges can be attributed to two main factors:

- **Lost in the middle phenomenon**. From a data perspective, natural language exhibits inherent biases, as people tend to prioritize important information at the beginning and end of contexts. As a result, both pre-training and instruction-tuning data may share such a trend. Under the supervision of the next token prediction task, Long-Context LLMs may also mimic this human tendency, overemphasizing the beginning and end tokens while sometimes ignoring the middle ones. This *lost in the middle* phenomenon was initially observed by Liu et al. [17], who revealed that Long-Context LLMs struggle to seek relevant information when the ground truth document is located in the middle of the input context.
- **Distraction issue**. From a model perspective, as the context window extends, an increasing number of tokens are involved in the self-attention mechanism, where attention scores for each query may dispersed across too many keys. In such a situation, keys with

* Co-corresponding Author. Email: shenzhy@pku.edu.cn
** Co-corresponding Author. Email: jwma@math.pku.edu.cn

different semantic meaning are more likely to overlap and become hard to distinguish by the query, making it difficult for the model to focus on relevant information. This *distraction* issue was first proposed by Tworkowski et al. [26].

To address the *lost in the middle* phenomenon, several studies utilize additional information from multi-document QA training data to generate augmented answers [33, 11], noticing that it is easy to acquire ground-truth documents when constructing training data. These augmented answers provide more supervision signals and can help the model to find relevant information within the middle of contexts. However, these methods may alter the answering pattern of the model. For instance, Yu [33] augmented the answers by adding a paraphrase of each relevant document, sometimes leading to verbose answers that do not align with human preferences.

To mitigate the *distraction issue*, a natural and straightforward approach is to reduce the number of input tokens, despite Long-Context LLMs theoretically supporting more tokens as input. This approach leads to retrieval-based methods, which typically retrieve the most relevant top-$k$ chunks or documents according to the query for input into LLMs. The performances of retrieval-based methods depend largely on the qualities of their retriever, and it has been manifested that Long-Context LLMs and retrieval-based methods have the potential to be combined to leverage the strengths of both [30]. However, in our early exploration (as shown in Section 3 and Table 1), we discover that a retriever with high or even 100% recall does not guarantee good performance in the downstream QA tasks. The low precision of retriever, which implies the inclusion of numerous irrelevant documents (referred as *distractors*) as input, can also lead to a significant degradation in QA performance.

**Table 1**: F1 scores of various input documents combinations on LongBench English multi-document QA datasets.

| Input | | Retriever | | HQA | 2WIKI | MSQ |
|---|---|---|---|---|---|---|
| Pos | Neg | Recall | Precision | | | |
| ✓ | ✓ | 100% | low | 54.79 | 51.03 | 35.54 |
| ✓ | × | 100% | 100% | **64.05** | **65.40** | **52.31** |
| × | ✓ | 0% | 0% | 26.82 | 22.54 | 8.36 |

In light of observations and analyses mentioned above, we propose an integrated Context Filtering Language Model (FltLM) aimed at enhancing the performance of Long-Context LLMs in multi-document QA tasks. FltLM is developed from vanilla Long-Context LLM with negligible number of introduced parameters, yet it manages to perform following two subtasks in order *in a single forward pass*: first, discriminating distractors and filtering them out via a soft mask mechanism; and second, comprehending or reasoning based on the remaining relevant documents to generate the answer. The former task is carried out by a context filter, while the latter is performed by the Long-Context LLM. Our main contributions are summarized as follows:

- We are the first to propose a context filter that can *automatically* identify all distractors based on the hidden text embedding of each document. In contrast, a typical retriever only outputs a sorted list of query-document relevance scores, requiring *manual* selection of input documents *case by case* using top-$k$ or top-$p$ strategies to ensure the best answer quality. The training objective of the context filter also encourages model to focus on documents at any position, thus alleviating the *lost in the middle* phenomenon.
- To filter out distractors, we design a soft mask mechanism that allows model to dynamically mask irrelevant tokens based on the discrimination results, which helps the model concentrate on relevant tokens and therefore mitigates the *distraction* issue. Moreover, this soft mask design makes the entire forward pass differentiable, enabling the joint end-to-end optimization of the context filter and the Long-Context LLM.
- Experimental results demonstrate that our proposed one-stage integrated FltLM significantly outperforms vanilla supervised fine-tuning and two-stage retrieval-based methods. By addressing the challenges and leveraging innovative solutions, FltLM seeks to redefine the capabilities of Long-Context LLMs in processing and understanding extensive and complex textual information.

## 2 Related work

### 2.1 Data-oriented methods

Data-oriented methods aim to address the *lost in the middle* phenomenon by constructing more informative supervising signals to strengthen the attention of the model. He et al. [11] proposed the Attention-Strengthening Multi-doc QA task, where labeled answers were organized in the order of question repetition, index prediction and answer summarization. This explicit extraction of the question and relevant document indices resembles the process of Chain-of-Thought [28], helping the model learn the reasoning pattern comprehensively during training. Yu [33] further augmented answers by adding paraphrases of relevant documents instead of solely predicting their indices. However, these approaches alter the answering pattern of the model and may result in verbose responses, even when the prompt does not ask to do so.

In accordance with the spirit of above methods, we also leverage additional information, i.e., the index of relevant documents to enhance long-context capabilities. However, our supervising signal is not presented in natural language form but rather as a list of 0/1 labels. As a consequence, our FltLM does not alter answering habits of the model. Meanwhile, under the guidance of context filter loss (defined in Section 4.2) rather than the language modeling loss, our model finds it easier to learn attention to documents at any position.

### 2.2 Retrieval-based methods

Retrieval-based methods employ a retriever to compute relevance scores between the query and all documents. Recent research mainly focus on dense retrieval [18, 3, 35, 27, 19, 29, 23], where a deep model learns text embeddings of query and documents and computes relevance scores with InfoNCE loss (or its variants) minimized.

The relevance score learned by InfoNCE loss turns out to be effective. However, retrieval-based methods face two challenges: first, in practice, top-$k$ strategy is usually adopted to get retrieval results. In this process, lots of distractors may be introduced to ensure a high recall rate, which may compromise the performance of downstream multi-document QA tasks. Second, it is inadequate to determine whether a document is relevant to the query by relying on the single value of the relevance score, since InfoNCE loss is *shift-invariant* (disscussed in Section 4.2), and only the differences between relevance scores are meaningful.

In our work, we solve the above issues by modifying the InfoNCE loss, enabling our model to function as a context filter capable of identifying and filtering out all distractors.

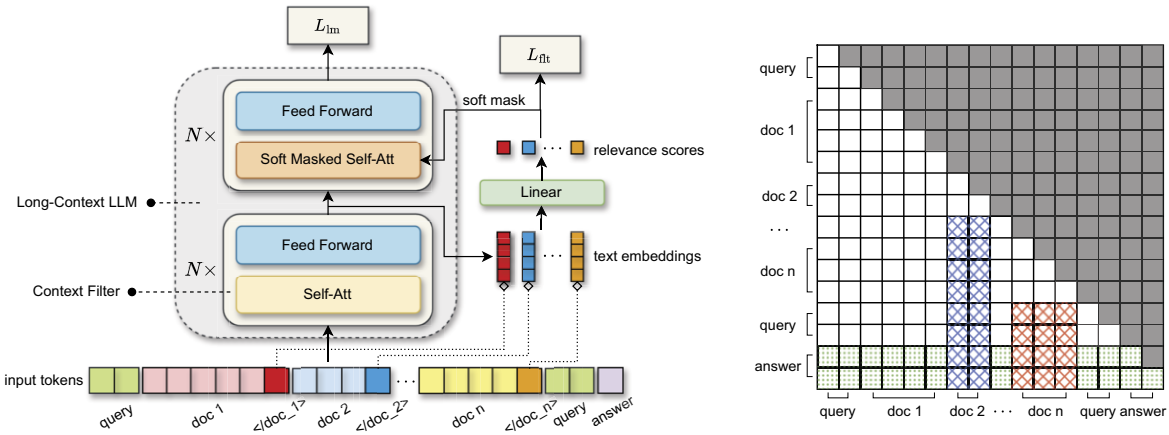



**Figure 1**: Overview. **Left:** Main architecture of FltLM. FltLM is built upon a $2N$-layer Long-Context LLM and integrates an $N$-layer context filter designed to compute relevance scores for each document. These scores are derived from text embeddings extracted from special input tokens </doc_i>, and are used to calculate the soft mask, which dynamically adjusts the self-attention mechanism in the last $N$ layers of the Long-Context LLM. **Right:** Soft mask mechanism. Solid gray-filled positions signify hard masks, while cross-hatched positions represent soft masks. Red and blue colors indicate different mask intensities, determined by the relevance scores from the context filter. Thanks to the soft mask mechanism, attention tends to be focused on relevant tokens during the inference stage, marked by dotted green positions, to generate high-quality answers.

## 3 Negative influence of distractors on multi-document QA task

In our early exploration, we examine the impact of relevant documents and distractors on the multi-document QA task. We select all LongBench [1] English multi-document QA datasets for experiments, using `chatglm3-6b-32k` [34] as the Long-Context LLM, which has reported the best performance on these datasets. Three combinations of input documents are tested, including all documents (Pos + Neg), relevant documents only (Pos) and distractors only (Neg). These combinations are designed to simulate an ordinary retriever with 100% recall and low precision, an oracle retriever with 100% recall and precision, and an incompetent retriever with 0% recall and precision respectively.

Results presented in Table 1 highlight that even when all relevant information is provided to the model, the introduction of numerous distractors in the input context significantly harms the performance on multi-document QA tasks, with degradations of $64.05\% \rightarrow 54.79\%, 65.40\% \rightarrow 51.03\%$ and $52.31\% \rightarrow 35.54\%$ across three datasets, respectively. Additionally, we evaluate model performance in a distractors-only setting, demonstrating that the model gains certain knowledge during pre-training and can correctly answers a subset of questions even in the absence of relevant documents. However, its performance is far from comparable to settings where relevant documents are available.

## 4 Methodology

In this section, we present our FltLM in detail. Figure 1(left) provides an overview of the main architecture of FltLM, wherein prompt tokens, normalization layers and residual connections have been omitted for clarity. The key insight of FltLM is straightforward: identifying all distractors and filtering them out to generate high-quality answers.

To elaborate, we assume a Long-Context LLM has $2N$ layers. During each forward pass, we utilize the initial $N$ layers of model to identify distractors detrimental to the answer. In this process, relevance scores are computed according to the semantic text embeddings output by the $N$-th layer. Based on these scores, a soft attention mask is applied to the last $N$ layers in an adaptive manner to mask out less informative tokens, concentrating the model on relevant documents to get better answers. Thanks to its integrated design, FltLM is able to function as both a context filter and a multi-document reader effectively.

### 4.1 *Extraction of semantic text embeddings for Long-Context LLMs*

Recently, LLM-based text embedding models are prevailing and exhibit state-of-the-art retrieval performance [19, 27, 23]. Different from BERT-style models characterized by bi-directional attention, they typically append a </s> token to the end of the text (either query or documents), and acquire its semantic vector by extracting the embedding from the last layer of </s>. Relevance scores $s_i$ are subsequently computed as the cosine similarity between the query embedding $h_q$ and the document embedding $h_{d_i}$:

$$\begin{aligned} h_q &= \text{LLM}(q\text{</s>})[-1],\ h_{d_i} = \text{LLM}(d_i\text{</s>})[-1] \\ s_i &= \langle h_q, h_{d_i} \rangle / \|h_q\| \|h_{d_i}\|. \end{aligned} \tag{1}$$

where $q$ and $d_i$ denotes the query and the $i$-th document respectively. For the reranker model that prioritizes accuracy over speed in ranking the input documents, relevance scores can be calculated as

$$\begin{aligned} \text{input} &= \text{"Query:}\{q\}\ \text{Document:}\{d_i\}\text{</s>"} \\ s_i &= \text{Linear}(\text{LLM}(\text{input})[-1]), \end{aligned} \tag{2}$$

where $\text{Linear}(\cdot)$ represents a linear layer utilized for regressing the relevance score based on the last layer embedding of the </s> token.

In our work, we extend Eqn. (2) to encode multiple documents simultaneously, while keeping the input format of multi-document QA tasks to the largest extent. The only modification to the conventional QA input format involves appending a special token </doc_i> following each $i$-th document:

$$\begin{aligned} \text{input} = \text{"}&\{\text{prompt}\} \\ &\text{Question:}\{q\} \\ &\text{Document 1:}\{d_1\}\text{</doc\_1>} \\ &\text{Document 2:}\{d_2\}\text{</doc\_2>} \\ &\cdots \\ &\text{Document n:}\{d_n\}\text{</doc\_n>} \\ &\{\text{prompt}\} \\ &\text{Question:}\{q\} \\ &\text{Answer:}\{a\}\text{"} \\ s_i = \text{Linear}(&\text{LLM}(\text{input})[p_i]), \end{aligned} \tag{3}$$

where $p_i$ represents the position index of </doc_i>. During the training and inference stages, the placeholder $a$ is replaced with the labeled answer or an empty string, respectively. Additionally, we adopt query-aware contextualization [17], which places the query before and after the documents, to construct our input. We refer Eqn.(3) as the *naive* strategy for the extraction of text embeddings.

Compared to Eqn. (2), our naive strategy enables the model to capture richer contextual information. This enhancement is achieved by exposing not only the $i$-th document but also the previous $(i-1)$ documents, along with their corresponding special tokens, to the special token </doc_i>. Although these preceding tokens are not directly related to the $i$-th document, they contribute additional contrastive information that improves the organization of the (key, value) space of the model. This concept aligns with the approach discussed in [26]. Consequently, Eqn. (3) yields more representative and discriminative text embeddings for each document.

In addition to the naive strategy, we explore two alternative approaches for extracting text embeddings as part of our ablation studies:

- First, we extract the embedding for each document according to Eqn. (3), while applying extra attention masks to the context filter to force each document to be invisible by others. In this way, text embeddings are extracted *independently*.
- Second, we propose a setting in which </doc_i> serves as a proxy for the aggregated information of the first $i$ documents. Under this *accumulative* strategy, relevance scores are formulated as follows:

$$s_i = \text{Linear}(\text{LLM}(\text{input})[p_i] - \text{LLM}(\text{input})[p_{i-1}]). \tag{4}$$

For detailed comparisons of these strategies, please see Section 6.

## 4.2 Training loss of context filter

Most LLM-based retriever is trained under the guidance of the following InfoNCE loss:

$$\begin{aligned} L_{\text{InfoNCE}} &= -\log \frac{e^{s_p/\tau}}{e^{s_p/\tau} + \sum\limits_{i\in\text{Neg}} e^{s_i/\tau}} \\ &= \log\left(1 + \sum_{i\in\text{Neg}} e^{(s_i - s_p)/\tau}\right), \end{aligned} \tag{5}$$

where $p$ is the index of positive (relevant) document and Neg stands for indices of all negative (irrelevant) documents. $\tau$ is the temperature parameter.

However, the *shift-invariant* nature of InfoNCE loss dooms that relevance scores $\{s_1, s_2, \cdots, s_n\}$ share the same loss with $\{s_1 + c, s_2 + c, \cdots, s_n + c\}$, making it theoretically infeasible to establish a universal threshold value across various query-document pairs to determine whether a document is relevant to the query. To confront this challenge, we introduce an absolute threshold $s^*$, expecting that the learned $s_i < s^*$ if and only if the $i$-th document is irrelevant. We set $s^* = 0$ without loss of generality, and the InfoNCE loss can be modified by adding two regularization terms, $e^{-s_p/\tau}$ for positive document with score less than 0, and $\sum_{i\in\text{Neg}} e^{s_i/\tau}$ for negative documents with scores greater than 0, thus imposing large penalty on inaccurately scored documents:

$$\begin{aligned} &L^*_{\text{InfoNCE}} \\ &= \log\left(1 + \sum_{i\in\text{Neg}} e^{(s_i - s_p)/\tau} + e^{-s_p/\tau} + \sum_{i\in\text{Neg}} e^{s_i/\tau}\right) \\ &= \log\left(1 + e^{-s_p/\tau}\right) + \log\left(1 + \sum_{i\in\text{Neg}} e^{s_i/\tau}\right). \end{aligned} \tag{6}$$

We then extend Eqn. (6) to accommodate multi-hop QA tasks, where models are asked to reading across multiple positive documents to synthesis an answer. Additionally, considering that overlooking a positive document poses greater risks than the inclusion of extra distractors, we set a margin $m > 0$ to encourage relevance scores of all positive documents exceed $m$. Ultimately, our training loss for context filter is represented as follows:

$$L_{\text{flt}} = \log\left(1 + \sum_{i\in\text{Pos}} e^{-(s_i - m)/\tau}\right) + \log\left(1 + \sum_{i\in\text{Neg}} e^{s_i/\tau}\right), \tag{7}$$

which shares similar spirit with ZLPR loss proposed by Su et al. [24].

## 4.3 Soft mask mechanism

The concept of soft mask is relative to that of typical hard masks, which are implemented via adding $-\infty$ biases to the attention scores during the computation of self-attention. This hard mask operation makes specific tokens completely invisible to others. In contrast, we design a learnable soft mask mechanism to make this operation differentiable and thereby more adaptable. Specifically, we compute mask intensities $I_i$ for each document based on their relevance scores as follows:

$$I_i = \min\{0, ws_i + b\}, \tag{8}$$

where $w$ and $b$ are trainable parameters. We hypothesize that $w > 0$, a proposition supported by subsequent experimental results, indicating that mask intensity is positively correlated with the relevance score. We also introduce a bias $b$, allowing our model to learn to either mask less significant positive documents as well (in the case where $b < 0$), or merely mask highly significant distractors (in the case where $b > 0$).

For the last $N$ layers of the model, we augment original attention matrix $A$ by directly adding the computed intensities as follows:

$$A[u_i{:}\,,\ l_i{:}\,u_i] \mathrel{+}= I_i, \tag{9}$$

where $l_i$ and $u_i$ stand for the lower and upper index bounds of the $i$-th document, respectively. Figure 1(right) illustrates our soft mask mechanism, assuming that document 2 and $n$ are identified as distractors. In this way, we reduce the visibility of irrelevant information to subsequent tokens during answer generation, therefore enhancing the performance of multi-document QA.

**Table 2**: QA performance of various methods across multiple datasets.

| Methods | Experimental Settings | | | HQA | 2WIKI | MSQ | Avg. |
|---|---|---|---|---|---|---|---|
| | $L_{\text{lm}}$ | $\lambda L_{\text{flt}}$ | Soft Mask | | | | |
| Baseline | × | × | × | 54.79 | 51.03 | 35.54 | 47.12 |
| Baseline + Retrieval | × | × | × | 55.63 | 55.71 | 39.36 | 50.23 |
| SFT | ✓ | × | × | 63.72 | 78.73 | 53.28 | 65.24 |
| SFT + Retrieval | ✓ | × | × | 62.89 | 79.21 | 53.83 | 65.31 |
| FltLM (w/o soft mask) | ✓ | ✓ | × | <u>65.67</u> | **80.39** | <u>54.80</u> | <u>66.95</u> |
| FltLM | ✓ | ✓ | ✓ | **67.53** | <u>80.16</u> | **55.05** | **67.58** |

### 4.4 FltLM

The training loss of FltLM is a weighted summation of following two losses: the context filter loss $L_{\text{flt}}$ supervised by indices of ground-truth documents, and language modeling loss $L_{\text{lm}}$ supervised by labeled answers. Formally, it can be written as

$$L = L_{\text{lm}} + \lambda L_{\text{flt}}, \tag{10}$$

where $\lambda$ is a hyper-parameter to balance the learning of context filter and Long-Context LLM.

## 5 Experiments

### 5.1 Training data construction

We collect data from the training sets of following three multi-hop QA datasets: HotpotQA [32], 2WikiMultiHopQA [12] and MuSiQue [25], all of which are based on Wikipedia. To meet the requirements of training a Long-Context LLM that necessitates long input sequences, we replace all short paragraphs with the corresponding full articles from the Wikipedia dataset [8] by matching their titles. This dataset contains full Wikipedia articles that have been preprocessed to remove markdown formatting and unwanted sections. We successfully match a total of 86,882 training samples, including 19,329 for HotpotQA, 55,695 for 2WikiMultiHopQA, and 10,858 for MuSiQue. For each sample, we construct long input data using all relevant documents, and then progressively introduce distractors until the length approaches ~32k tokens. We shuffle the position of each document and exclude any samples that exceed the maximum input length or computational constraints of our devices. The final training set comprises 84,762 samples.

### 5.2 Experimental settings

**Training schemes**. We initialize our Long-Context LLM with the `chatglm3-6b-32k` [34] checkpoint, renowned for its strong performance and state-of-the-art results on the LongBench [1], a comprehensive benchmark for long-context understanding. We choose the *naive* strategy for text embeddings extraction unless specifically stated. For hyper-parameters, we set $\lambda = 0.5$ and designate $\tau$ as learnable. Our model is trained using LoRA [13] with a rank of $r = 16$, a dropout ratio of $p = 0.1$, and $\alpha = 64$, employing data-distributed parallel training [15] with a total batch size of 32 and training epoch of 1. The maximum learning rates are set to $1 \times 10^{-4}$ for all LoRA modules and $1 \times 10^{-2}$ for $w$, $b$, and the linear layer that computes $s_i$, following a linear decay schedule with a warm-up ratio of 0.01. To reduce GPU memory usage, we apply techniques such as Flash Attention v2 [7], mixed precision training [20], and gradient checkpointing [5]. All experiments can run on 4 × 80G Nvidia A800 GPUs.

**Evaluation for QA performance**. We evaluate our model using the three English multi-document QA datasets include in LongBench, specifically referred to as HQA, 2WIKI, and MSQ. These datasets are derived from the testing sets of HotpotQA, 2WikiMultiHopQA, and MuSiQue respectively, and we have verified that there is no overlap of questions between these datasets and our training set. In line with the evaluation metrics of LongBench, we use the N-gram based F1-score to measure the quality of generated answers.

**Evaluation for context filter**. We recover the ground-truth documents for HQA, 2WIKI, and MSQ by exactly matching each sample with its corresponding entry in the original datasets. Based on these matches, we evaluate the performance of our context filter through metrics such as recall, precision, and F1-score, providing a comprehensive analysis of its efficacy in filtering irrelevant contextual information.

### 5.3 Main results and comparisons

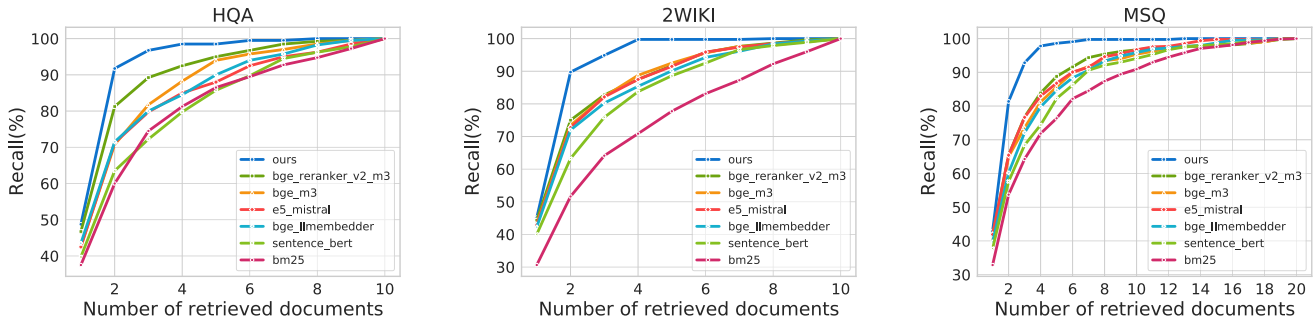
**The effectiveness of FltLM**.

To evaluate the effectiveness of our FltLM, we compare it against several mainstream solutions for multi-document QA tasks, including: (i) utilizing a general Long-Context LLM (Baseline); (ii) fine-tuning the Long-Context LLM with labeled data (SFT); and (iii) retrieval-based methods combined with the aforementioned two models. For document retrieval, we employ a state-of-the-art retriever, BGE-reranker-v2-m3 [3], to fetch the top-$k$ documents to ensure a high recall rate of 95%.

Our main results, depicted in Table 2, showcase the QA performance of various methods. Notably, the best results are highlighted in **bold**, with the second best results <u>underlined</u>. Examining Table 2, it is evident that our FltLM significantly outperforms supervised fine-tuning and retrieval-based methods. On average, it achieves substantial improvements of 2.34% (65.24% → 67.58%) and 2.31% (65.31% → 67.58%) respectively, and these enhancements are consistent across all datasets.

The optimized parameters, $w = 0.289$ and $b = -0.206$, align with our expectation that $w > 0$. This result indicates that soft masks should be applied to both irrelevant documents and marginally relevant ones.

To further validate the effectiveness of each component within FltLM, we also conduct experiments on a version of FltLM devoid of the soft mask. Even without this feature, it also achieves higher F1-score compared to supervised fine-tuning (65.24% → 66.95%). We attribute these improvements to two main factors. First, the additional term $\lambda L_{\text{flt}}$ involves with labels from ground-truth documents, providing extra supervising signals that boost our model to learn more knowledge. Second, as discussed in Section 1, the intrinsic bias of natural language may allow Long-Context LLMs to neglect mid-text contents while still performing well on the next token prediction task. However, the distractors prediction task introduced by $\lambda L_{\text{flt}}$ requires a comprehensive understanding of all documents. Consequently, our




(a) HQA dataset. (b) 2WIKI dataset. (c) MSQ dataset.

**Figure 2**: Recall of different retrievers across multiple datasets.

FltLM is able to learn better semantic features that benefit the downstream tasks.

On the other hand, the incorporation of the soft mask mechanism also improves our model, evidenced by an increase in the QA F1-score of $0.63\%$ ($66.95\% \rightarrow 67.58\%$). This mechanism effectively filters out distractors, allowing our FltLM to concentrate more on relevant information and thereby mitigating the *distraction* issue. Furthermore, we highlight that this modification is risk-free since $w$ and $b$ are learnable. Setting $w = b = 0$ and keeping them fixed can directly degrade FltLM to its variant that lacks the soft mask mechanism.

**Table 3**: Impact of input document order on QA performance.

| | Doc. Order | HQA | 2WIKI | MSQ | Avg. |
|---|---|---|---|---|---|
| Baseline | original | 54.79 | 51.03 | 35.54 | 47.12 |
| | reordered | 58.16 | 52.27 | 42.11 | 50.85 (+3.73) |
| SFT | original | 63.72 | 78.73 | 53.28 | 65.24 |
| | reordered | 65.09 | 80.78 | 57.25 | 67.71 (+2.46) |
| FltLM | original | 67.53 | 80.16 | 55.05 | 67.58 |
| | reordered | 67.31 | 79.26 | 59.46 | 68.68 (**+1.10**) |

**Analysis of the *lost in the middle* phenomenon**. To further confirm that FltLM alleviates the lost in the middle phenomenon, we conduct experiments by reordering the input documents. Specifically, we move half of the relevant documents to the beginning of the input and the other half to the end, while maintaining relative positions of the remaining documents. If the lost in the middle phenomenon does not exist, this reordering will not affect the model's performance. However, as shown in Table 3, the average F1-score of the baseline model increases by $3.73\%$ ($47.12\% \rightarrow 50.85\%$), highlighting a severe lost in the middle problem. In contrast, the average F1-score of the SFT model increases by $2.46\%$ ($65.24\% \rightarrow 67.71\%$), while that of FltLM increases by only $1.10\%$ ($67.58\% \rightarrow 68.68\%$). These results suggest that our model effectively strengthens attentions to the middle relevant, making it more robust to document order compared to the SFT model.

**Table 4**: Attention scores for positive and negative documents, averaged across samples ($\times 10^{-2}$).

| | Positive Doc. | Negative Doc. |
|---|---|---|
| SFT | 3.54 | 0.31 |
| FltLM | 2.61 | 1.19 |

**Analysis of the *distraction* issue**.

To evaluate how effectively FltLM addresses the issue of distraction, we analyze the model's attention scores during the generation of the first token of the answer. The token-level attention scores are calculated by averaging across multiple attention heads in the final $N$ layers. Document-level scores are then obtained by summing the relevant token-level scores. Table 4 presents the average attention scores for both positive and negative documents. The results indicate that FltLM, enhanced by the soft mask mechanism, is more focused on relevant contents compared to the SFT model.

**The effectiveness of context filter**. As an ancillary benefit, in the training process of FltLM, we yield a context filter designed to identify all distractors within the long input context. It is noteworthy that this context filter can also function as a conventional dense retriever by generating a sorted list of relevance scores. This capability prompts a natural question: how effectively can our context filter perform retrieval tasks? To explore this, we benchmark it against several existing retrievers, including BM25, Sentence-BERT [22], and BGE-llmembedder [35], as well as three state-of-the-art ones: BGE-m3 [3], BGE-reranker-v2-m3 [3], and E5-Mistral [27]. Figure 2 illustrates how recall varies with the number of documents retained for different retrievers. It is universally observed across different datasets that our context filter surpasses all the aforementioned retrievers, achieving the saturation of recall at the fastest rate.

## 6 Ablation studies

**FltLM v.s. two-stage filter-and-then-read strategy**. As an integrated model, FltLM can perform distractors identification and QA tasks through a single forward pass, while achieving strong performance. However, is this integrated end-to-end design necessary? To answer this question, we propose and evaluate a two-stage filter-and-then-read strategy as a comparison. Specifically, we train a context filter and concurrently fine-tune the Long-Context LLM under the guidance of $L_{\text{flt}}$ and $L_{\text{lm}}$, respectively. During the inference stage, the context filter first calculates relevance scores, and documents with $s_i > 0$ are then selected and fed into the fine-tuned Long-Context LLM for further processing.

**Table 5**: FltLM v.s. filter-and-then-read strategy.

| Methods | HQA | 2WIKI | MSQ | Avg. |
|---|---|---|---|---|
| SFT | 63.72 | 78.73 | <u>53.28</u> | <u>65.24</u> |
| Filter-and-then-read | <u>64.48</u> | <u>79.09</u> | 46.99 | 63.52 |
| FltLM | **67.53** | **80.16** | **55.05** | **67.58** |

Table 5 provides evaluation results for this two-stage strategy, revealing that it is an intuitive yet less effective method, with an average decline of $-4.06\%$ ($67.58\% \rightarrow 63.52\%$) compared to the one-stage FltLM. We attribute this performance degradation to two main

**Table 6**: Impact of different text embedding extraction strategies on QA performance. Our naive extraction strategy results in minimal QA performance degradations, largely maintaining original capabilities of the Long-Context LLM.

| Extraction Strategies | Experimental Settings $L_{\text{lm}}$ | $L_{\text{flt}}$ | HQA | 2WIKI | MSQ | Avg. |
|---|---|---|---|---|---|---|
| None (Baseline) | × | × | **54.79** | **51.03** | 35.54 | **47.12** |
| Independent | × | ✓ | 47.96 | 40.35 | 32.32 | 40.21 |
| Accumulative | × | ✓ | 20.34 | 5.65 | 12.67 | 12.89 |
| Naive (Ours) | × | ✓ | 50.80 | 49.11 | **35.98** | 45.30 |

factors. First, although our context filter shows potential, it remains imperfect and may fail to retrieve all related information to answer the question. For instance, in the HQA and 2WIKI datasets, where the context filter performs relatively well, this strategy does improve answer quality compared to supervised fine-tuning. However, in the MSQ dataset, where the context filter exhibits poor recall and F1-score, the model frequently fails to collect sufficient relevant contexts, resulting in suboptimal answers. Moreover, the separate training of context filter and Long-Context LLM prevents the potential reciprocal benefits of combining their respective loss functions, a topic we will discuss at the end of this section.

**Comparisons of different text embeddings extraction strategies**. In our study, we adopt the *naive* approach to derive text embeddings for the $i$-th document, specifically by extracting the hidden vector of </doc_i>, similar to BGE-landmark [18]. However, this method presents a potential issue since previous documents are also associated with this special token, raising doubts on its adequacy in capturing the unique semantic features of the $i$-th document. To address this concern and validate our approach, we also implement two alternative strategies introduced in section 4.1 to train our context filter. Meanwhile, as a baseline, we experiment with a *pairwise* training strategy as well, where each forward pass computes relevance score for a single document using Eqn. (2).

**Table 7**: Comparison of context filter metrics across different text embedding extraction strategies.

| Extraction Strategies | Precision | Recall | F1-score |
|---|---|---|---|
| Pairwise | 91.42 | 83.70 | 85.35 |
| Independent | 89.76 | 83.93 | 84.61 |
| Accumulative | 93.15 | 86.86 | 88.30 |
| Naive (Ours) | **93.51** | **88.04** | **89.09** |

Table 7 summarizes the average context filter metrics of various methods we explored. Our initial naive approach turns out to be the most effective, followed by the accumulative strategy. In contrast, strategy that applies hard masks and extracts embeddings independently tend to yield slightly poorer outcomes compared to the pairwise training baseline. Notably, strategies that are capable of capturing richer contextual information, namely the naive and accumulative ones, significantly outperform their counterparts, underscoring their ability to produce highly representative and discriminative text embeddings for each document. These results could provide valuable insights and potentially inspire improvements in text embedding models.

We further examine the compatibility of various text embedding extraction strategies with our ultimate goal of multi-document QA. To this end, we freeze the last $N$ layers of the Long-Context LLM and fine-tune it solely under the guidance of $L_{\text{flt}}$. Our findings, presented in Table 6, reveal that the application of the naive strategy results in only minimal reductions in QA performance even without the guidance of $L_{\text{lm}}$. This observation suggests that the latent features of the LLM could potentially perform additional tasks beyond next token prediction, while largely retaining its original functionalities.

**The impact of $\lambda$ on the performance of FltLM**. As discussed in section 4.4, we introduce the hyper-parameter $\lambda$ to control the trade-off between the learning processes of the context filter and the Long-Context LLM. In pursuit of our objective to enhance the capabilities Long-Context LLM, we treat $L_{\text{flt}}$ as an auxiliary loss and roughly set $\lambda = 0.5$. To further analyze the impact of $\lambda$ on the performance of FltLM, we conduct ablation studies with $\lambda = 0.2$ and $1.0$.

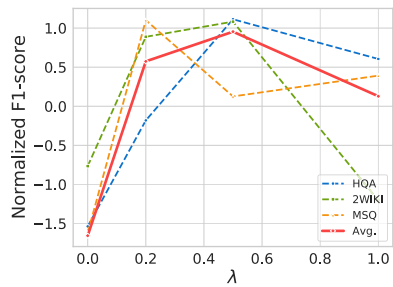



**Figure 3**: QA performance of FltLM across different values of $\lambda$.

Figure 3 describes how QA performance vaires with the value of $\lambda$, where $\lambda = 0$ represents standard supervised fine-tuning. Metrics for each dataset are normalized independently to visualize the variation trends on the same scale. Figure 3 reveals that FltLM consistently achieves higher F1-scores than supervised fine-tuning at $\lambda = 0.2$ and $0.5$, highlighting the robustness of our method. However, performance decreases when $\lambda$ is raised to $1.0$. An optimal balance is achieved at $\lambda = 0.5$, yielding the best average results for our FltLM.

## 7 Conclusion

In this paper, we propose FltLM, a novel integrated Long-Context LLM that significantly enhances multi-document QA performance, addressing the two critical challenges of *the lost in the middle phenomenon* and the *distraction* issue. FltLM employs a context filter with a soft mask mechanism which identifies and dynamically excludes the less relevant content, thereby focusing on the essential information for improved long-context understanding. By embedding the context filter directly within the architecture of the model, FltLM not only streamlines computational processes to a single forward pass but also markedly surpasses supervised fine-tuning and retrieval-based methods in complex QA settings.

The emergence of FltLM opens up new avenues for advanced natural language processing applications. Future work will focus on optimizing the context filtering process, extending the applicability of the model to other natural language processing tasks such as sophisticated document summarization and in-depth content generation, and integrating emerging neural network paradigms to further enhance performance and scalability. This progression promises to improve the capabilities of Long-Context LLMs significantly, making them more versatile and effective across various domains.

## Acknowledgements

This work was supported by the Natural Science Foundation of China under grant 62071171 and the high-performance computing platform of Peking University.

## References

[1] Y. Bai, X. Lv, J. Zhang, H. Lyu, J. Tang, Z. Huang, Z. Du, X. Liu, A. Zeng, L. Hou, Y. Dong, J. Tang, and J. Li. Longbench: A bilingual, multitask benchmark for long context understanding, 2023.

[2] Y. Chang, K. Lo, T. Goyal, and M. Iyyer. Booookscore: A systematic exploration of book-length summarization in the era of llms. In The Twelfth International Conference on Learning Representations, 2023.

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

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

[5] T. Chen, B. Xu, C. Zhang, and C. Guestrin. Training deep nets with sublinear memory cost. arXiv preprint arXiv:1604.06174, 2016.

[6] Y. Chen, S. Qian, H. Tang, X. Lai, Z. Liu, S. Han, and J. Jia. Longlora: Efficient fine-tuning of long-context large language models. In The Twelfth International Conference on Learning Representations, 2023.

[7] T. Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In The Twelfth International Conference on Learning Representations, 2023.

[8] W. Foundation. Wikimedia downloads. URL https://dumps.wikimedia.org.

[9] gkamradt. Llmtest_needleinahaystack. https://github.com/gkamradt/LLMTest_NeedleInAHaystack, 2023.

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

[11] J. He, K. Pan, X. Dong, Z. Song, Y. Liu, Y. Liang, H. Wang, Q. Sun, Z. Songxin, X. Zejian, et al. Never lost in the middle: Improving large language models via attention strengthening question answering. arXiv preprint arXiv:2311.09198, 2023.

[12] X. Ho, A.-K. Duong Nguyen, S. Sugawara, and A. Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In D. Scott, N. Bel, and C. Zong, editors, Proceedings of the 28th International Conference on Computational Linguistics, pages 6609–6625, Barcelona, Spain (Online), Dec. 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580.

[13] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[14] J. Li, M. Wang, Z. Zheng, and M. Zhang. Loogle: Can long-context language models understand long contexts? arXiv preprint arXiv:2311.04939, 2023.

[15] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania, et al. Pytorch distributed: experiences on accelerating data parallel training. Proceedings of the VLDB Endowment, 13(12):3005–3018, 2020.

[16] S. Li, F. Xue, C. Baranwal, Y. Li, and Y. You. Sequence parallelism: Long sequence training from system perspective. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2391–2404, 2023.

[17] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics, 12:157–173, 2024.

[18] K. Luo, Z. Liu, S. Xiao, and K. Liu. Bge landmark embedding: A chunking-free embedding method for retrieval augmented long-context large language models. arXiv preprint arXiv:2402.11573, 2024.

[19] X. Ma, L. Wang, N. Yang, F. Wei, and J. Lin. Fine-tuning llama for multi-stage text retrieval. arXiv preprint arXiv:2310.08319, 2023.

[20] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, et al. Mixed precision training. In International Conference on Learning Representations, 2018.

[21] B. Peng, J. Quesnelle, H. Fan, and E. Shippole. Yarn: Efficient context window extension of large language models. In The Twelfth International Conference on Learning Representations, 2023.

[22] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3982–3992, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410.

[23] J. M. Springer, S. Kotha, D. Fried, G. Neubig, and A. Raghunathan. Repetition improves language model embeddings. arXiv preprint arXiv:2402.15449, 2024.

[24] J. Su, M. Zhu, A. Murtadha, S. Pan, B. Wen, and Y. Liu. Zlpr: A novel loss for multi-label classification. arXiv preprint arXiv:2208.02955, 2022.

[25] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. MuSiQue: Multihop questions via single-hop question composition. Transactions of the Association for Computational Linguistics, 2022.

[26] S. Tworkowski, K. Staniszewski, M. Pacek, Y. Wu, H. Michalewski, and P. Miłoś. Focused transformer: Contrastive training for context scaling. Advances in Neural Information Processing Systems, 36, 2024.

[27] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei. Improving text embeddings with large language models. arXiv preprint arXiv:2401.00368, 2023.

[28] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35:24824–24837, 2022.

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

[30] P. Xu, W. Ping, X. Wu, L. McAfee, C. Zhu, Z. Liu, S. Subramanian, E. Bakhturina, M. Shoeybi, and B. Catanzaro. Retrieval meets long context large language models. In The Twelfth International Conference on Learning Representations, 2023.

[31] J. Yang. Longqlora: Efficient and effective method to extend context length of large language models. arXiv preprint arXiv:2311.04879, 2023.

[32] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Conference on Empirical Methods in Natural Language Processing (EMNLP), 2018.

[33] Y. Yu. "paraphrasing the original text" makes high accuracy long-context qa. arXiv preprint arXiv:2312.11193, 2023.

[34] A. Zeng, X. Liu, Z. Du, Z. Wang, H. Lai, M. Ding, Z. Yang, Y. Xu, W. Zheng, X. Xia, et al. Glm-130b: An open bilingual pre-trained model. In The Eleventh International Conference on Learning Representations, 2022.

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

[36] P. Zhang, Z. Liu, S. Xiao, N. Shao, Q. Ye, and Z. Dou. Soaring from 4k to 400k: Extending llm's context with activation beacon. arXiv preprint arXiv:2401.03462, 2024.

[37] D. Zhu, N. Yang, L. Wang, Y. Song, W. Wu, F. Wei, and S. Li. Pose: Efficient context window extension of llms via positional skip-wise training. In The Twelfth International Conference on Learning Representations, 2023.