BLOCK-ATTENTION FOR EFFICIENT RAG

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce Block-Attention, an attention mechanism designed to address the increased inference latency and cost in Retrieval-Augmented Generation (RAG) scenarios. Traditional approaches often encode the entire context. Instead, Block-Attention divides retrieved documents into discrete blocks, with each block independently calculating key-value (KV) states except for the final block. In RAG scenarios, by defining each passage as a block, Block-Attention enables us to reuse the KV states of passages that have been seen before, thereby significantly reducing the latency and the computation overhead during inference. The implementation of Block-Attention involves block segmentation, position re-encoding, and fine-tuning the LLM to adapt to the Block-Attention mechanism. Experiments on four RAG benchmarks demonstrate that after block fine-tuning, the Block-Attention model achieves performance comparable to self-attention models (68.4% vs 67.9% on Llama3) or even superior performance (62.8% vs 59.6% on Mistral). Notably, Block-Attention significantly reduces the time to first token (TTFT) and floating point operations (FLOPs) to a very low level. It only takes 45 ms to output the first token for an input sequence with a total length of 32K. Compared to the self-attention models, the time consumption and corresponding FLOPs are reduced by 98.7% and 99.8%, respectively.

1 INTRODUCTION

Retrieval-Augmented Generation (RAG) (Li et al., 2022) is a crucial technology for mitigating knowledge hallucination in Large Language Models (LLMs). By utilizing retrieval technology, LLMs can seamlessly access passages stored in external databases, grounding their responses in the content of these passages. To the best of our understanding, RAG has emerged as the most effective method for infusing specific domain knowledge into LLMs in real-world scenarios.

However, everything has two sides, and RAG is no exception. Generally, for each user query, in order to ensure that a passage with the "correct knowledge" is successfully recalled, it is a common practice to retrieve multiple passages—typically between 5 to 30 in most scenarios (Kwiatkowski et al., 2019; Joshi et al., 2017). These passages are then incorporated into the input prompt for the LLM. As a result, the inference efficiency decreases significantly due to the increased sequence length of this extended input prompt. Specifically, the inference latency, measured as the time to first token (TTFT), is considerably higher for a RAG-LLM compared to a non-RAG LLM (Li et al., 2023; Zhu et al., 2024).

Given that the passages in the external databases might have been computed, it is natural to restore their KV cache for fast inference and avoid re-computing these passages. Nonetheless, for autoregressive LLMs, the KV states are inherently context-dependent, which means the KV states for the same passage will vary in different contexts. As a result, when encountering an unseen query, the model must undertake a complete re-encoding of the KV states to ensure an accurate prediction.

In this paper, we propose the Block-Attention method, which reduces the TTFT and computation FLOPs of RAG-LLMs to a level nearly equivalent to that of non-RAG LLMs, while fully maintaining the same accuracy level. As shown in Figure 1, the main idea of Block-Attention is to divide the entire input sequence into several blocks. Each block independently calculates its KV states through self-attention, without any attention to other blocks. Only the final block is able to attend other blocks, *i.e.,* the user query is able to attend all the retrieved documents in the previous blocks.

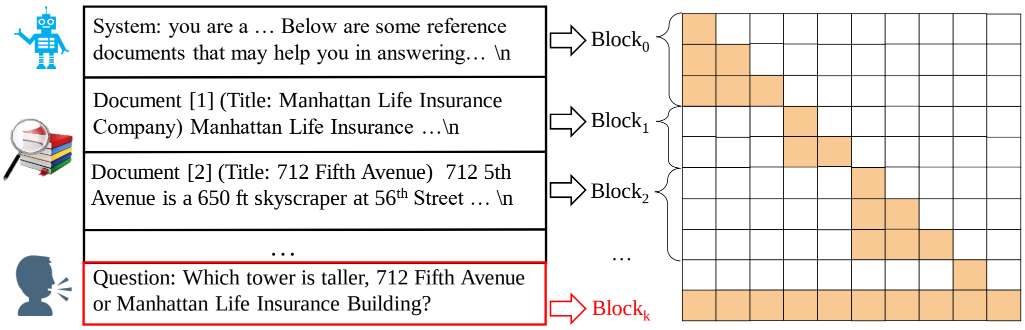

Figure 1: The Block-Attention Masks

When utilizing Block-Attention in RAG scenarios, we may achieve substantial benefits by defining each passage as a single block and caching its KV states in the memory for further reuse.

The implementation of Block-Attention can be easily achieved through the following steps: 1) Encode all blocks except the last one separately; 2) calculating the positional encoding for each based on its position in the input prompt; 3) Integrating these blocks to compute the KV states for the final block. However, the primary challenge in utilizing Block-Attention is that the LLMs have not been exposed to such an attention mechanism during their training, leading to difficulties in accurately interpreting the input prompt. In our preliminary experiments, we attempted to directly implement Block-Attention in the LLMs without updating any parameters. Unfortunately, this approach resulted in a substantial decrease in performance, with the average accuracy of Llama3-8B on four RAG benchmarks falling from 67.9% to 48.0%.

To address this challenge, we implemented a fine-tuning process for the LLMs to adapt to the Block-Attention mechanism. Our experiments demonstrated that, after approximately 100 - 1000 fine-tuning steps, the Block-Attention model achieved a full recovery of its original accuracy across both in-domain and out-domain scenarios, impressively increasing from 48.0% to 68.4%. This outcome underscores the Block-Attention LLMs' capability to uphold inference accuracy while significantly enhancing inference efficiency in RAG scenarios.

We evaluate Block-Attention on four RAG benchmarks. Experimental results demonstrate that after fine-tuning, the average accuracy of the Block-Attention model on the benchmarks can remain comparable to (68.4% vs 67.9% on Llama3) or even slightly exceed (62.3% vs 59.6% on Mistral) the performance of self-attention models. In terms of efficiency, we counted the TTFT and FLOPs to the first token (FLOPs-TFT) of the block-attention model when the length of user input is 50 and the total length of the input sequence gradually increases. We found that the longer the total length, the more obvious the improvement of block-attention on efficiency. When the length of the input sequence reaches 32K, the TTFT and FLOPs-TFT of the block-attention model are 1.3% and 0.2% of that of the self-attention model, respectively.[1]

2 Block-Attention

2.1 Main Idea

Let $\mathcal{S} = \{s_0, s_1, ..., s_n\}$ represents the input sequence, where each s represents a token. We denote the KV states associated with $\mathcal{S}$ as $\mathcal{K} = \{k_0, k_1, ..., k_n\}$ and $\mathcal{V} = \{v_0, v_1, ..., v_n\}$, respectively. For an auto-regressive model Θ_{LLM}, since the computation of the KV states is dependent on their preceding tokens, when a text block $\{s_i, ..., s_j\}$ changes to a new text block $\{s'_i, ..., s'_m\}$, for the new sequence $\mathcal{S}' = \{s_0, ..., s'_i, ..., s'_m, s_{j+1}, ..., s_n\}$, its KV states become $\mathcal{K}' = \{k_0, ..., k'_i, ..., k'_m, k'_{j+1}, ..., k'_n\}$ and $\mathcal{V}' = \{v_0, ..., v'_i, ..., v'_m, v'_{j+1}, ..., v'_n\}$. It is evident that although only one block $\{s_i, ..., s_j\}$ has changed, due to the auto-regressive nature of LLMs, the KV states of all subsequent blocks must be re-encoded.

[1]Given that the KV cache is already a mature and low-cost technology (Qin et al., 2024; Lan et al., 2023; Lee et al., 2021), in this paper we do not take the cost of KV cache into account.

In the scenario of RAG, a given passage frequently recurs in the retrieval results for numerous queries (Qin et al., 2024), making it a natural consideration to explore the possibility of reusing its KV states. Unfortunately, the auto-regressive nature of the encoding approach, as previously outlined, imposes constraints that render the KV states nearly non-reusable across distinct queries. Our research is dedicated to examining a novel attention mechanism, Block-Attention, that designed to necessitate the re-computation of only the altered text blocks between two input sequences, thereby achieving an outcome equivalent to that of full sequence re-encoding.

As illustrated in Figure 1, the essence of Block-Attention is to divide the input sequence $\mathcal{S}$ into several independent blocks. Each block autonomously calculates its KV states through self-attention, without considering other blocks. The final block, however, has the unique capability to integrate information from preceding blocks. A primary advantage of this method is the modular independence it provides: when a block b_i is updated to b'_i, re-encoding only the KV states of the affected block $k_{b'_i}$, $v_{b'_i}$, and those of the final block k_{b_k}, v_{b_k}, is sufficient to obtain the updated sequence's KV states.

To develop a Block-Attention LLM capable of precise inference, we must tackle three key challenges:

- How do we segment blocks?
- How should the positional encoding be calculated for each block?
- How can the LLM be adapted to the Block-Attention mechanism?

These issues will be addressed in detail in Sections 2.2, 2.3, and 2.4, respectively.

2.2 Block Segmentation

The primary principle of block division is to segment semantically independent parts of the prompt into separate blocks. In RAG scenarios, since the retrieved passages are originally mutually independent, it is natural to divide them into different blocks. Therefore, let's go back to the left part of Figure 1, where we allocate each passage to a single block and designate the user's query as the final block. This principle extends to other scenarios as well. For example, in the context of code generation tasks, a function may be treated as one block; in multi-turn dialogues, each turn could be segmented into an individual block. In this paper, our primary focus is on the application of Block-Attention in RAG, with the exploration of other scenarios reserved for future research.

2.3 Position Re-Encoding

The second problem is to re-encoding the positional information. Although the same passage may appear in multiple input prompts, its position generally varies. Therefore, when we attempt to reuse the KV states of a block, we need to re-encode its positional information. The process of re-encoding is simple and straightforward: taking the rotary positional encoding (RoPE) as an example, assume we wish to change the positional encoding of a block $b = \{s_i, ..., s_j\}$ to $b' = \{s_{i_\Delta}, ..., s_{j_\Delta}\}$, then we only need three steps:

1) For the token s_i, its positional encoding vector $f(s_i, i)$ is calculated using the following formula:

$$f(x_i, i) = \begin{pmatrix} \cos i\theta_0 & -\sin i\theta & \cdots & 0 & 0 \\ \sin i\theta_0 & \cos i\theta & \cdots & 0 & 0 \\ 0 & 0 & \cdots & \cos i\theta_{\frac{d}{2}-1} & -\sin i\theta_{\frac{d}{2}-1} \\ 0 & 0 & \cdots & \sin i\theta_{\frac{d}{2}-1} & \cos i\theta_{\frac{d}{2}-1} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ \cdots \\ p_{d-1} \end{pmatrix} \tag{1}$$

2) We rotating x_i counterclockwise by $i\theta$ degrees, to re-set its positional encoding to zero:

$$f(x_i, 0) = \begin{pmatrix} \cos i\theta_0 & \sin i\theta & \cdots & 0 & 0 \\ -\sin i\theta_0 & \cos i\theta & \cdots & 0 & 0 \\ 0 & 0 & \cdots & \cos i\theta_{\frac{d}{2}-1} & \sin i\theta_{\frac{d}{2}-1} \\ 0 & 0 & \cdots & -\sin i\theta_{\frac{d}{2}-1} & \cos i\theta_{\frac{d}{2}-1} \end{pmatrix} f(x_i, i) \tag{2}$$

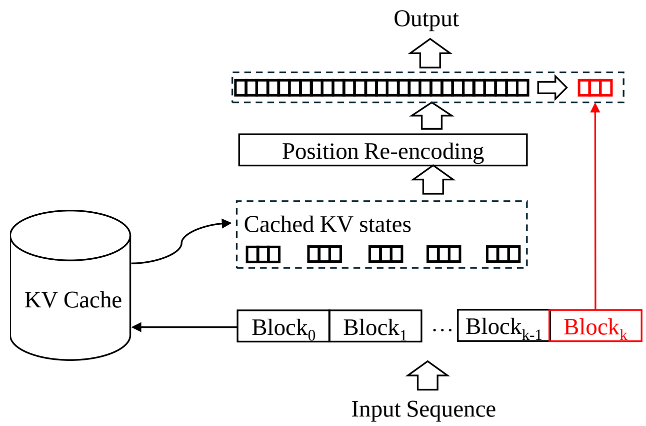

Figure 2: The Inference Pipeline of Block-Attention Model

3) Then, by performing a clockwise rotation of $(i_\triangle)\theta$ degrees, we obtain the final positional encoding vector:

$$f(x_{i_\triangle}, i_\triangle) = f(x_i, i_\triangle)$$
$$\begin{pmatrix} \cos{(i_\triangle)\theta_0} & -\sin{(i_\triangle)\theta} & \cdots & 0 & 0 \\ \sin{(i_\triangle)\theta_0} & \cos{(i_\triangle)\theta} & \cdots & 0 & 0 \\ 0 & 0 & \cdots & \cos{(i_\triangle)\theta_{\frac{d}{2}-1}} & -\sin{(i_\triangle)\theta_{\frac{d}{2}-1}} \\ 0 & 0 & \cdots & \sin{(i_\triangle)\theta_{\frac{d}{2}-1}} & \cos{(i_\triangle)\theta_{\frac{d}{2}-1}} \end{pmatrix} f(x_i, 0) \tag{3}$$

For the remaining tokens within block b, namely $s_{i+1}, \ldots, s_j$, we can re-encode their positional information in a similar manner. Although the formulas presented above may seem complex, the principle is quite straightforward: **first set the positional encoding to zero, and then rotate it to the updated position**. One might wonder why we do not simply rotate by $(i_\triangle - i)\theta$ degrees directly? The reason is to mitigate the potential for errors in updating positional encodings within practical applications: in the KV cache, the positional encoding of the initial token of each block is standardized to zero, and with only the updated positional index $i_\triangle$, we can readily determine their new positional encoding vectors as per Equation 3.

2.4 Block Fine-tune

Due to the LLM's reliance on full self-attention during the training phase, a direct switch to Block-Attention during inference might result in a significant discrepancy between the training and inference states. Our preliminary findings indicate that introducing Block-Attention without subsequent fine-tuning could precipitate a substantial decrease in performance, with the average accuracy dropping significantly from 67.9% to 48.0%. Adapting the LLM to Block-Attention through fine-tuning, which we refer to as "**block fine-tune**," is quite straightforward. The only difference from the standard SFT process is the modification of the traditional lower-triangular attention mask matrix to the attention mask matrix depicted in the right part of Figure 1. With this masking matrix, tokens in all blocks except the last are restricted to attending only to information within their own block, thereby ensuring consistency between training and inference.

2.5 Inference

In inference, the Block-Attention model retrieves the KV states of blocks from the KV cache and concatenates them into the complete input KV states. The detailed process of the inference stage is depicted in Figure 2. Initially, we query and extract the KV states of the first $k-1$ blocks from the cache. Then, based on the position of each block within the input sequence, we calculate their positional encoding using Equation 3. Finally, using the KV states of the first $k-1$ blocks, we compute the KV states of the last block as well as the model output. In the RAG scenarios, the last block is the user query.

Input Prompt

You are an intelligent AI assistant. Please answer questions based on the user's instructions. Below are some reference documents that may help you in answering the user's question. \n

Doc-1

-Title: Manhattan Life Insurance Company \n
Manhattan Life Insurance Company, incorporated on May 29, 1850, is a life……\n

Doc-2~n

- Title: New York Life Building \n
The New York Life Insurance Building, New York, located at 51 Madison Avenue, Manhattan, New York City, across……\n

…

User Query

Please write a high-quality answer for the given question using only the provided search documents (some of which might be irrelevant).
Question: Which tower is taller, 712 Fifth Avenue or Manhattan Life Insurance Building?

Output

The 712 Fifth Avenue tower

Figure 3: The Inference Pipeline of Block-Attention Model. The retrieved documents at the top have the highest relevance to the user query.

3 Experiments

After the above analysis, there exists two concerns about the Block-Attention method: 1) Can the Block-Attention model achieve the same accuracy as self-attention in RAG scenarios? 2) How much can the Block-Attention mechanism improve the efficiency? The following experimental results will reveal the answers to these two questions. In Sections 3.5, we analyze the accuracy of Block-Attention models. Meanwhile, in Section 3.6, we demonstrate the efficiency of Block-Attention.

3.1 Datasets

Train Dataset We construct the training datasets by randomly sampling 20,000 instances from these TriviaQA (TQA) (Joshi et al., 2017) and 2WikiMultiHopQA (2Wiki) (Ho et al., 2020) for fine-tuning models. Each training sample consists of (1) a question, (2) 10 passages retrieved from these two datasets using the Contriver toolkit[2], which identifies the 10 most relevant passages, and (3) an answer generated by GPT-4 based on the retrieved passages. The reason to use the GPT-4 answer instead of the ground-truth answers is because the answer might not be present in our retrieved passages. This discrepancy could lead the model to overlook the content of the retrieved passages and generate outputs directly.

Evaluation Dataset We evaluate the performance of our proposed Block-Attention mechanism and baseline models on four widely-used RAG benchmarks: Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (TQA) (Joshi et al., 2017), HotpotQA (HQA) (Yang et al., 2018), and 2Wiki-MultiHopQA (2Wiki) (Ho et al., 2020). Since the training samples include instances from TQA and 2Wiki, their test sets are considered in-domain, whereas HQA and NQ are treated as out-of-domain test sets. Following Kandpal et al. (2023) and Liu et al. (2024), we use accuracy as our primary evaluation metric, judging whether any correct answers appear in the predicted output. To mitigate biases arising from output length, we set a maximum token limit of 200 for the output sequences.

3.2 Input Format

The format of input prompt for all datasets follows Liu et al. (2024). For retrieved passages, we concatenate them in ascending order of retrieval score. An example is shown in Figure 3.

[2]https://github.com/facebookresearch/contriever

3.3 Base Model & Baselines

We implement the Block-Attention mechanism on two base language models: Llama3-8B-base[3] and Mistral-7B-v3.0[4]. After applying block fine-tuning, these models are denoted as *Llama3-block-ft* and *Mistral-block-ft*. For comparison, we implemented three baselines for each of these two base language models.

- **Strong baselines:** *Llama3-vanilla-sft* and *Mistral-vanilla-sft* are established by fine-tuning the base language models on our constructed training dataset. These models serve as the upper bound performance for the Block-Attention models. We aim for the Block-Attention model's performance to approach these as closely as possible.
- **Weak baselines:** *Llama3-block-w/o-ft* and *Mistral-block-w/o-ft*. These models transition the attention mechanism of *Llama3-vanilla-sft* and *Mistral-vanilla-sft* to a Block-Attention mechanism **without any block fine-tuning**. The outcomes of these models represent the lower bounds for the Block-Attention model's effectiveness, given that they have not undergone any adaptation to Block-Attention during the training phase.
- **w/o-pos:** *Llama3-block-ft-w/o-pos* and *Mistral-block-ft-w/o-pos* are baselines that also fine-tuned to adapt the Block-Attention mechanism, while no additional position re-encoding operations described in Section 2.3 are conducted. This group of models will be used to evaluate the effectiveness of the proposed position re-encoding process.

Please note that we have not utilized instruction-tuning models as our base language model, such as Llama3-instruct. The reason is our inability to entirely re-implement the fine-tuning processes of these models, which precludes a direct comparison between self-attention and Block-Attention methodologies under identical experimental conditions. Instead, we perform vanilla fine-tuning and block fine-tuning on the same base models using the same dataset, ensuring a fair comparison.

Additionally, our idea has some similarities with PCW (Ratner et al., 2023b) and Prompt Cache (Gim et al., 2024). However, the focus of PCW is on extending the context window rather than efficient inference. When we directly applied their attention mechanism in the RAG scenario, their accuracy was even lower than the weak baselines, so we did not choose it as a baseline. We also re-implemented Prompt Cache. Then, we had an interesting finding: their results are exactly the same as the two weak baselines: *Llama3-block-w/o-ft* and *Mistral-block-w/o-ft*. Therefore, for the results of Prompt Cache, please directly refer to the result of these two models.[5]

3.4 Setup

All experiments are done with 8 NVIDIA A800 GPUs, we set the learning rate $\alpha = 2 \times 10^{-5}$, batch size $b = 64$ and the number of epochs $n = 1$ for all models. We use a linear learning rate warmup for the first 20 steps. Its training is in bfloat16 format, using DeepSpeed ZeRO Stage-2[6] and Flash-Attention V2 (Dao et al., 2022). The Flash-Attention toolkit is used for efficient inference for our models and baselines.

3.5 Main Results

From the results in Table 1, we can draw three important conclusions:

1) It is not advisable to directly switch from self-attention to Block-Attention, as it will lead to a sharp accuracy drop. For example, removing the Block fine-tune process leads to the 21.99% average absolute performance decrease on all four benchmarks for Llama3-8B and Mistral-7B models.

2) However, if we use the Block-Attention mechanism in the fine-tuning stage, then the resultant model has almost the same performance as the self-attention model or is even slightly better on

[3]https://huggingface.co/meta-llama/Meta-Llama-3-8B

[4]https://huggingface.co/mistralai/Mistral-7B-v0.3

[5]Please note that in addition to accuracy, Prompt Cache also has serious efficiency issues when applied in the RAG field: once the rank of a passage in the input sequence changes, its KV cache cannot be reused.

[6]`https://github.com/microsoft/DeepSpeed`

Models	TQA	2wiki	NQ	HQA	In-Dom. Avg.	Out-Dom. Avg.	Mix Avg.
Llama3-vanilla-sft	72.8	**73.4**	55.7	**69.5**	**73.9**	**62.0**	67.9
Llama3-block-ft	**73.0**	73.3	**56.2**	68.5	73.6	63.2	**68.4**
Llama3-block-ft-w/o-pos	70.8	66.8	52.9	61.4	69.3	58.5	63.9
Llama3-block-w/o-ft	62.7	48.1	43.3	39.6	54.7	41.4	48.0
Mistral-vanilla-sft	66.6	69.5	46.0	56.4	68.3	50.9	59.6
Mistral-block-ft	**69.9**	**70.5**	**50.8**	**57.9**	**70.2**	**54.4**	**62.3**
Mistral-block-ft-w/o-pos	67.8	67.4	47.5	52.4	67.3	50.4	58.8
Mistral-block-w/o-ft	51.9	44.6	32.1	27.8	47.7	29.7	38.7

Table 1: Accuracy of different models on four benchmarks. Please note that the "average" column does not depict a simple arithmetic mean of the preceding columns, as the number of samples contained in each test set varies.

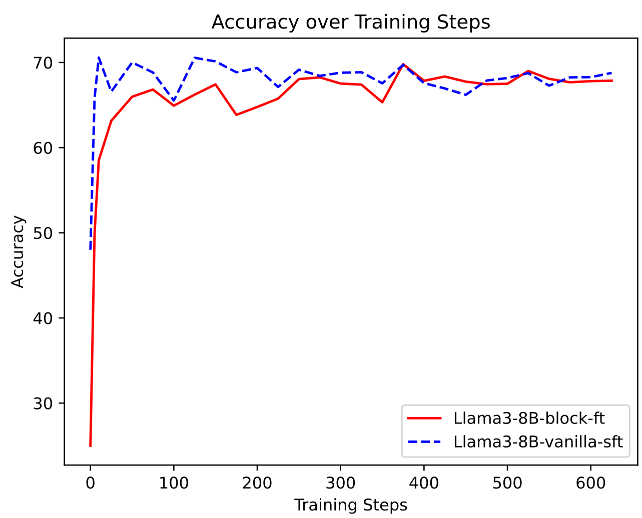

Figure 4: The accuracy of model checkpoint at different steps

some datasets. For instance, *Mistral-block-ft* outperforms the strong baseline *Mistral-vanilla-sft*, with 2.7% absolute improvements on four benchmarks.

3) The position re-encoding operations are essential for the Block-Attention Model. Removing it leads to significant performance drop—average 4% decreases in accuracy on four datasets.

The first conclusion is easy to understand: The model has never seen an input sequence in the Block-Attention manner during the training process. Therefore, the sharp drop makes sense. Next, we will focus on the second conclusion. From Table 1, we can easily find that the models trained with two attention methods, namely *Llama3-vanilla-sft* and *Llama3-block-ft*, not only have comparable effects on in-domain test sets but also have little difference on out-domain test sets. This indicates that in the RAG scenario, it is completely feasible to replace self-attention with Block-Attention, and there will be almost no performance loss.

Moreover, the experimental results on Mistral show some interesting results: the accuracy of Block-Attention models actually slightly outperformed the results of self-attention models on most datasets. It raised a conjecture: is Block-Attention a better attention mechanism than the self-attention mechanism? Intuitively, the semantics of each retrieved passage are mutually independent, and the self-attention method may cause the representation of the passage to be interfered by the context and not accurate enough. However, we still need sufficient experiments to verify the correctness of this assertion, so this is only a conjecture and not our conclusion.

Prompt Length	50	512	1K	2K	4K	8K	16K	32K
TTFT-vanilla	26	50	87	167	330	691	1515	3638
TTFT-block	26	26(48%)	26(71%)	26(84%)	27(91%)	29(95%)	34(97%)	45(98.7%)
FLOPs-TFT-vanilla	7.5e+11	7.6e+12	1.5e+13	3.0e+13	6.1e+13	1.2e+14	2.45e+14	4.9e+14
FLOPs-TFT-block	7.5e+11	7.5e+11	7.5e+11	7.5e+11	7.5e+11	7.5e+11	7.5e+11	7.5e+11
Reduction	-	90.1%	95.0%	97.5%	98.7%	99.3%	99.6%	99.8%

Table 2: The Time and FLOPs consumed by the first token of a user input with a length of 50 tokens under different total length of the retrieved passages

Finally, one may still be interested in knowing exactly how many training steps are needed for the model to adapt to the Block-Attention mechanism. Therefore, we counted the accuracy of the Llama3-8B model on the four test sets at different fine-tuning steps and plotted it in Figure 4. It can be observed that at the beginning stage of fine-tuning, there is a huge performance difference between the two models. It makes sense because the model needs more training steps to adapt to the Block-Attention manner. After about 400 training steps, the model completely adapts to the new attention mechanism, and the two models show comparable accuracy.

3.6 Inference Efficiency of Block-Attention

In the previous section, we already addressed our first concern: After fine-tuning, the Block-Attention model can achieve similar or even better performance than the self-attention model. In this section, we focus on our second concern: How much can the Block-Attention mechanism reduce the TTFT and FLOPs-TFT?

To quantify the effects of the Block-Attention mechanism on the efficiency, we show in Table 2 the respective TTFTs and FLOPs-TFT of the *Llama3-block-ft* and *Llama3-vanilla-sft* when the KV states of all retrieved passages have been pre-computed and cached in memory. Obviously, the acceleration effect is gratifying: Once the length of the input sequence is 512 and the length of user input is 50, using Block-Attention can reduce TTFT by 48% and reduce FLOPs-TFT by 90.1%. As the total length increases, the TTFT and FLOPs-TTF of the Block-Attention model maintain an essentially unchanged trend. When the length reaches 32K, the acceleration effect reaches an astonishing 98.7%, and the consumption of FLOPs-TFT is even reduced by 99.8%. We may simply conclude the results as: *with greater text comes greater necessity for Block-Attention*.

3.7 Discussion

From the experimental results, we can figure out the effects of Block-Attention on existing RAG applications: Under the existing technical paradigm, developers need to deliberately balance the trade-off between accuracy and latency. Therefore, they have to limit the number of retrieved passages to a certain number, such as 3 to 10. With Block-Attention, the impact of the number of retrieved passages on inference speed will be greatly reduced, which will empower the developers to freely choose the number of retrieved passages that gives the optimal effect without any hesitation.

Furthermore, some dynamic RAG paradigms such as ReAct (Yao et al., 2023) have been proven to enhance the general task-solving ability of the LLM. However, compared to vanilla RAG, such paradigms need to iteratively compute the KV states of retrieved passages for single user input, which results in a disastrous inference latency. With the help of Block-Attention, the additional latency consumption in this aspect can be reduced to a level that users can tolerate. Developers will be able to safely use such paradigms in online scenarios to optimize the reasoning ability of LLM.

4 Related Works

4.1 Retrieval-Augmented Generation (RAG)

RAG is a widely used technique to improve generations of language models by using retrieved nearest-neighbor documents or passages as references, which typically involves two stages: retrieval and generation. Before generation, retrieval finds most similar passages with the user query or the context, by using BM25 or dense retrieval model (Lee et al., 2021; Lan et al., 2023; 2024). After collecting retrieved passages, there are numerous techniques to incorporate the knowledge during generation. Earlier works includes concatenation (Izacard & Grave, 2021) and cross-attention (Borgeaud et al., 2022). Some studies have also begun to attempt to directly use the retriever as text generator, that is, text generation is performed by selecting context-aware phrases from a collection of supporting documents. (Lan et al., 2023; Cao et al., 2024)

Recently, LLMs becomes the most powerful paradigm for most NLP tasks, and simply concatenating all retrieved documents into the context of LLMs becomes the most simple and effective way for retrieval-augmented generation (RAG). For example, Self-RAG (Asai et al., 2023) leverage a critic model to decide which content in retrieved passages should be used during generation. As a specific application of RAG, tool learning is widely used to call external APIs to retrieve related passages from external database or tools to solve knowledge-intensive tasks (Schick et al., 2023).

4.2 EFFICIENCY OF RAG

The efficieny of LLM inference gradually decreases as the sequence length of the concatenation of retrieved passages rapidly increases, leading to a huge challenges for retrieval augmented generation of LLMs. To solve this problem, there are numerous novel architectures, model quantization and inference strategies are proposed. For example, the linear attention and Mixture of Expert (Fedus et al., 2022) architecture are widely used architectures for reducing the complexity of LLM inference (Shazeer, 2019; Zhang & Cai, 2022; Fedus et al., 2022). Speculative decoding leverages the parallel of language models to speed up the inference (Leviathan et al., 2023). However, most previous works are not designed for RAG scenarios. Recently, RAG Cache(Jin et al., 2024) caches the intermediate states of retrieved knowledge in GPUs, showing 4x TTFT.

Another stream of researches focus on individually and parallely process each documents, which is related to our work, such as FiD (Izacard & Grave, 2021), PCW (Ratner et al., 2023a) and Prompt Cache (Gim et al., 2024). However, FiD is widely used for encode-decoder architecture, and not compatible for the decoder-only architecture of LLMs, since they need to concatenate the hidden states for decoder during inference. PCW focuses on extending the context window rather than efficient inference. Prompt Cache enables the efficient inference for RAG applications by designing the Prompt Markup Language (PML), while their pre-computed KV cache of retrieved passages are bound to the specific IDs, leading to its limited applications. In other words, once the rank of a passage in the input sequence changes, its KV cache cannot be reused.

5 CONCLUSION

We introduce Block-Attention to optimize the inference efficiency of LLM in RAG scenarios. Its essence lies in separately calculating the KV states of independent blocks in the input prompt. Block-Attention enables us to pre-compute the KV states for all passages and cache them in memory, thus avoiding repeated computation of the KV states of the same passage during inference.

Our experimental results across various RAG benchmarks demonstrated the profound impact of Block-Attention. We showed that Block-Attention can maintain the original reasoning accuracy of LLM while significantly reducing its TTFT and FLOPs-TTF in RAG scenarios. The effectiveness of Block-Attention becomes increasingly apparent as the number of passages increases or the frequency of retrievals increases. When considering the implementation of Block-Attention in your applications, bear in mind this guiding principle: With greater text comes greater necessity for Block-Attention.

We also want to note that Block-Attention plays an important role in many scenarios and is not limited to RAG only. We only introduced its effects on RAG because of that, due to some confidentiality reasons, we cannot disclose how we use it in our industrial applications. We look forward to researchers from the community being able to further explore the potential of Block-Attention and apply it to more appropriate scenarios.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection, 2023. URL https://arxiv.org/abs/2310.11511.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens, 2022. URL https://arxiv.org/abs/2112.04426.

Bowen Cao, Deng Cai, Leyang Cui, Xuxin Cheng, Wei Bi, Yuexian Zou, and Shuming Shi. Retrieval is accurate generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oXYZJXDdo7.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022. URL https://arxiv.org/abs/2205.14135.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2022. URL https://arxiv.org/abs/2101.03961.

In Gim, Guojun Chen, Seung seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt cache: Modular attention reuse for low-latency inference, 2024. URL https://arxiv.org/abs/2311.04934.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 874–880, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.74. URL https://aclanthology.org/2021.eacl-main.74.

Chao Jin, Zili Zhang, Xuanlin Jiang, Fangyue Liu, Xin Liu, Xuanzhe Liu, and Xin Jin. Ragcache: Efficient knowledge caching for retrieval-augmented generation, 2024. URL https://arxiv.org/abs/2404.12457.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge, 2023. URL https://arxiv.org/abs/2211.08411.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav

Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026.

Tian Lan, Deng Cai, Yan Wang, Heyan Huang, and Xian-Ling Mao. Copy is all you need, 2023. URL https://arxiv.org/abs/2307.06962.

Tian Lan, Deng Cai, Yan Wang, Yixuan Su, Heyan Huang, and Xian-Ling Mao. Exploring dense retrieval for dialogue response selection. *ACM Trans. Inf. Syst.*, 42(3), jan 2024. ISSN 1046-8188. doi: 10.1145/3632750. URL https://doi.org/10.1145/3632750.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. Learning dense representations of phrases at scale, 2021. URL https://arxiv.org/abs/2012.12624.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding, 2023. URL https://arxiv.org/abs/2211.17192.

Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. A survey on retrieval-augmented text generation, 2022. URL https://arxiv.org/abs/2202.01110.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. Compressing context to enhance inference efficiency of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6342–6353, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.391. URL https://aclanthology.org/2023.emnlp-main.391.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9.

Ruoyu Qin, Zheming Li, Weiran He, Mingxing Zhang, Yongwei Wu, Weimin Zheng, and Xinran Xu. Mooncake: A kvcache-centric disaggregated architecture for llm serving, 2024. URL https://arxiv.org/abs/2407.00079.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. Parallel context windows for large language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6383–6402, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.352. URL https://aclanthology.org/2023.acl-long.352.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. Parallel context windows for large language models, 2023b. URL https://arxiv.org/abs/2212.10947.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Yacmpz84TH.

Noam Shazeer. Fast transformer decoding: One write-head is all you need, 2019. URL https://arxiv.org/abs/1911.02150.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018. URL https://arxiv.org/abs/1809.09600.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

Yizhe Zhang and Deng Cai. Linearizing transformer with key-value memory. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 346–359, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.24. URL `https://aclanthology.org/2022.emnlp-main.24`.

Yun Zhu, Jia-Chen Gu, Caitlin Sikora, Ho Ko, Yinxiao Liu, Chu-Cheng Lin, Lei Shu, Liangchen Luo, Lei Meng, Bang Liu, and Jindong Chen. Accelerating inference of retrieval-augmented generation via sparse context selection, 2024. URL `https://arxiv.org/abs/2405.16178`.

A A CASE STUDY FROM HARRY POTTER

System Instruction (Block$_0$) : You are an intelligent AI assistant. Please answer questions based on the user's instructions. Below are some reference documents that may help you in answering the user's question.

Retrieved Passages: **Block 1:** Title: session 1. Harry rose early to pack the next day; the Hogwarts Express would be leaving an hour after the funeral. Downstairs, he found the mood in the Great Hall subdued. Everybody was wearing their dress robes and no one seemed very hungry. Professor McGonagall had left the thronelike chair in the middle of the staff table empty. Hagrid's chair was deserted too; Harry thought that perhaps he had not been able to face breakfast, but Snape's place had been unceremoniously filled by Rufus Scrimgeour. Harry avoided his yellowish eyes as they scanned the Hall; Harry had the uncomfortable feeling that Scrimgeour was looking for him. Among Scrimgeour's entourage Harry spotted the red hair and horn-rimmed glasses of Percy Weasley. Ron gave no sign that he was aware of Percy, apart from stabbing pieces of kipper with unwonted venom.

Over at the Slytherin table Crabbe and Goyle were muttering together. Hulking boys though they were, they looked oddly lonely without the tall, pale figure of Malfoy between them, bossing them around. Harry had not spared Malfoy much thought. His animosity was all for Snape, but he had not forgotten the fear in Malfoy's voice on that tower top, nor the fact that he had lowered his wand before the other Death Eaters arrived. Harry did not believe that Malfoy would have killed Dumbledore. He despised Malfoy still for his infatuation with the Dark Arts, but now the tiniest drop of pity mingled with his dislike. Where, Harry wondered, was Malfoy now, and what was Voldemort making him do under threat of killing him and his parents?

Block 2: Title: session 2. Harry's thoughts were interrupted by a nudge in the ribs from Ginny. Professor McGonagall had risen to her feet, and the mournful hum in the Hall died away at once.

"It is nearly time," she said. "Please follow your Heads of Houses out into the grounds. Gryffindors, after me." They filed out from behind their benches in near silence. Harry glimpsed Slughorn at the head of the Slytherin column, wearing magnificent, long, emerald green robes embroidered with silver. He had never seen Professor Sprout, Head of the Hufflepuffs, looking so clean; there was not a single patch on her hat, and when they reached the entrance hall, they found Madam Pince standing beside Filch, she in a thick black veil that fell to her knees, he in an ancient black suit and tie reeking of mothballs.

Block 3: Title: session 3. They were heading, as Harry saw when he stepped out onto the stone steps from the front doors, toward the lake. The warmth of the sun caressed his face as they followed Professor McGonagall in silence to the place where hundreds of chairs had been set out in rows. An aisle ran down the center of them: There was a marble table standing at the front, all chairs facing it. It was the most beautiful summer's day. An extraordinary assortment of people had already settled into half of the chairs; shabby and smart, old and young. Most Harry did not recognize, but a few he did, including members of the Order of the Phoenix: Kingsley Shacklebolt; Mad-Eye Moody; Tonks, her hair miraculously returned to vividest pink; Remus Lupin, with whom she seemed to be holding hands; Mr. and Mrs. Weasley; Bill supported by Fleur and followed by Fred and George, who were wearing jackets of black dragon skin. Then there was Madame Maxime, who took up two and a half chairs on her own; Tom, the landlord of the Leaky Cauldron in London; Arabella Figg, Harry's Squib neighbor; the hairy bass player from the Wizarding group the Weird Sisters; Ernie Prang, driver of the Knight Bus; Madam Malkin, of the robe shop in Diagon Alley; and some people whom Harry merely knew by sight, such as the barman of the Hog's Head and the witch who pushed the trolley on the Hogwarts Express. The castle ghosts were there too, barely visible in the bright sunlight, discernible only when they moved, shimmering insubstantially on the gleaming air.

Block 4: Title: session 4. Harry, Ron, Hermione, and Ginny filed into seats at the end of a row beside the lake. People were whispering to each other; it sounded like a breeze in the grass, but the birdsong was louder by far. The crowd continued to swell; with a great rush of affection for both of them, Harry saw Neville being helped into a seat by Luna. Neville and Luna alone of the D.A. had responded to Hermione's summons the night that Dumbledore had died, and Harry knew why: They were the ones who had missed the D.A. most . . . probably the ones who had checked their coins regularly in the hope that there would be another meeting.

Block 5: Title: session 5. fury, Dolores Umbridge, an unconvincing expression of grief upon her toadlike face, a black velvet bow set atop her iron-colored curls. At the sight of the centaur Firenze, who was standing like a sentinel near the water's edge, she gave a start and scurried hastily into a seat a good distance away. The staff was seated at last. Harry could see Scrimgeour looking grave and dignified in the front row with Professor McGonagall. He wondered whether Scrimgeour or any of these important people were really sorry that Dumbledore was dead. But then he heard music, strange, otherworldly music, and he forgot his dislike of the Ministry in looking around for the source of it. He was not the only one: Many heads were turning, searching, a little alarmed.

"In there," whispered Ginny in Harry's ear."'

Block 6: Title: session 6. And he saw them in the clear green sunlit water, inches below the surface, reminding him horribly of the Inferi: a chorus of merpeople singing in a strange language he did not understand, their pallid faces rippling, their purplish hair flowing all around them. The music made the hair on Harry's neck stand up, and yet it was not unpleasant. It spoke very clearly of loss and of despair. As he looked down into the wild faces of the singers, he had the feeling that they, at least, were sorry for Dumbledore's passing. Then Ginny nudged him again and he looked around.

Hagrid was walking slowly up the aisle between the chairs. He was crying quite silently, his face gleaming with tears, and in his arms, wrapped in purple velvet spangled with golden stars, was what Harry knew to be Dumbledore's body. A sharp pain rose in Harry's throat at this sight: For a moment, the strange music and the knowledge that Dumbledore's body was so close seemed to take all warmth from the day. Ron looked white and shocked. Tears were falling thick and fast into both Ginny's and Hermione's laps.

They could not see clearly what was happening at the front. Hagrid seemed to have placed the body carefully upon the table. Now he retreated down the aisle, blowing his nose with loud trumpeting noises that drew scandalized looks from some, including, Harry saw, Dolores Umbridge . . . but Harry knew that Dumbledore would not have cared. He tried to make a friendly gesture to Hagrid as he passed, but Hagrid's eyes were so swollen it was a wonder he could see where he was going. Harry glanced at the back row to which Hagrid was heading and realized what was guiding him, for there, dressed in a jacket and trousers each the size of a small marquee, was the giant Grawp, his great ugly boulderlike head bowed, docile, almost human. Hagrid sat down next to his half-brother, and Grawp patted Hagrid hard on the head, so that his chair legs sank into the ground. Harry had a wonderful momentary urge to laugh. But then the music stopped, and he turned to face the front again.

Block 7: Title: session 7. A little tufty-haired man in plain black robes had got to his feet and stood now in front of Dumbledore's body. Harry could not hear what he was saying. Odd words floated back to them over the hundreds of heads. "Nobility of spirit" . . . "intellectual contribution" . . . "greatness of heart" . . . It did not mean very much. It had little to do with Dumbledore as Harry had known him. He suddenly remembered Dumbledore's idea of a few words, "nitwit," "oddment," "blubber," and "tweak," and again had to suppress a grin. . . . What was the matter with him?

There was a soft splashing noise to his left and he saw that the merpeople had broken the surface to listen too. He remembered Dumbledore crouching at the water's edge two years ago, very close to where Harry now sat, and conversing in Mermish with the Merchieftainess. Harry wondered where Dumbledore had learned Mermish. There was so much he had never asked him, so much he should have said. . . .

And then, without warning, it swept over him, the dreadful truth, more completely and undeniably than it had until now. Dumbledore was dead, gone. . . . He clutched the cold locket in his hand so tightly that it hurt, but he could not prevent hot tears spilling from his eyes: He looked away from Ginny and the others and stared out over the lake, toward the forest, as the little man in black droned on. . . . There was movement among the trees. The centaurs had come to pay their respects too. They did not move into the open but Harry saw them standing quite still, half hidden in shadow, watching the wizards, their bows hanging at their sides. And Harry remembered his first nightmarish trip into the forest, the first time he had ever encountered the thing that was then Voldemort, and how he had faced him, and how he and Dumbledore had discussed fighting a losing battle not long thereafter. It was important, Dumbledore said, to fight, and fight again, and keep fighting, for only then could evil be kept at bay, though never quite eradicated. . . .

And Harry saw very clearly as he sat there under the hot sun how people who cared about him had stood in front of him one by one, his mother, his father, his godfather, and finally Dumbledore, all determined to protect him; but now that was over. He could not let anybody else stand between him and Voldemort; he must abandon forever the illusion he ought to have lost at the age of one, that the shelter of a parent's arms meant that nothing could hurt him. There was no waking from his nightmare, no comforting whisper in the dark that he was safe really, that it was all in his imagination; the last and greatest of his protectors had died, and he was more alone than he had ever been before.

Block 8: Title: session 8. The little man in black had stopped speaking at last and resumed his seat. Harry waited for somebody else to get to their feet; he expected speeches, probably from the Minister, but nobody moved.

Then several people screamed. Bright, white flames had erupted around Dumbledore's body and the table upon which it lay: Higher and higher they rose, obscuring the body. White smoke spiraled into the air and made strange shapes: Harry thought, for one heart-stopping moment, that he saw a phoenix fly joyfully into the blue, but next second the fire had vanished. In its place was a white marble tomb, encasing Dumbledore's body and the table on which he had rested.

There were a few more cries of shock as a shower of arrows soared through the air, but they fell far short of the crowd. It was, Harry knew, the centaurs' tribute: He saw them turn tail and disappear back into the cool trees. Likewise, the merpeople sank slowly back into the green water and were lost from view.

Harry looked at Ginny, Ron, and Hermione: Ron's face was screwed up as though the sunlight were blinding him. Hermione's face was glazed with tears, but Ginny was no longer crying. She met Harry's gaze with the same hard, blazing look that he had seen when she had hugged him after winning the Quidditch Cup in his absence, and he knew that at that moment they understood each other perfectly, and that when he told her what he was going to do now, she would not say, "Be careful," or "Don't do it," but accept his decision, because she would not have expected anything less of him. And so he steeled himself to say what he had known he must say ever since Dumbledore had died.

Block 9: Title: session 9. "Ginny, listen . . ." he said very quietly, as the buzz of conversation grew louder around them and people began to get to their feet, "I can't be involved with you anymore. We've got to stop seeing each other. We can't be together."

She said, with an oddly twisted smile, "It's for some stupid, noble reason, isn't it?"

"It's been like . . . like something out of someone else's life, these last few weeks with you," said Harry. "But I can't . . . we can't . . . I've got things to do alone now."

She did not cry, she simply looked at him.

"Voldemort uses people his enemies are close to. He's already used you as bait once, and that was just because you're my best friend's sister. Think how much danger you'll be in if we keep this up. He'll know, he'll find out. He'll try and get to me through you."

Block 10: Title: session 10. "What if I don't care?" said Ginny fiercely.

"I care," said Harry. "How do you think I'd feel if this was your funeral . . . and it was my fault. . . ."

She looked away from him, over the lake.

“I never really gave up on you,” she said. “Not really. I always hoped. . . . Hermione told me to get on with life, maybe go out with some other people, relax a bit around you, because I never used to be able to talk if you were in the room, remember? And she thought you might take a bit more notice if I was a bit more — myself.” “Smart girl, that Hermione,” said Harry, trying to smile. “I just wish I’d asked you sooner. We could’ve had ages . . . months . . . years maybe. . . .”

“But you’ve been too busy saving the Wizarding world,” said Ginny, half laughing. “Well . . . I can’t say I’m surprised. I knew this would happen in the end. I knew you wouldn’t be happy unless you were hunting Voldemort. Maybe that’s why I like you so much.”

User Question 1: What will Harry Potter’s next actions be after the funeral?

Output of Llama3-block-ft: Based on the provided search documents, it appears that Harry Potter’s next actions will be to continue his mission to defeat Lord Voldemort. This is evident from the following quote:

”And Harry saw very clearly as he sat there under the hot sun how people who cared about him had stood in front of him one by one, his mother, his father, his godfather, and finally Dumbledore, all determined to protect him; but now that was over. He could not let anyone else stand between him and Voldemort; he must abandon forever the illusion he ought to have lost at the age of one, that the shelter of a parent’s arms meant that nothing could hurt him.”

This quote suggests that Harry has accepted the reality of Dumbledore’s death and is now prepared to take matters into his own hands, continuing his mission to defeat Voldemort.

Output of Llama3-vanilla-sft: According to the provided documents, after the funeral, Harry Potter will continue to pursue his mission to defeat Voldemort. This is evident from his statement Ï’ve got things to do alone nowänd his subsequent declaration that he cannot be involved with Ginny anymore due to the dangers that would ensue if Voldemort were to target her as a way to get to him.

Additionally, Harry’s thoughts and feelings during the funeral suggest that he has accepted the reality of Dumbledore’s death and is prepared to carry on without him. This is evident from his statement Ï can’t be involved with you anymore. We’ve got to stop seeing each other. We can’t be together.ẅhich implies that he is ready to focus on his mission and put aside his personal relationships.

Overall, it appears that Harry’s next actions will involve continuing his quest to defeat Voldemort, possibly alone, and putting aside his personal relationships in order to focus on his mission.

Output of Llama3-block-w/o-ft: Based on the provided search documents, it appears that Harry Potter’s next actions will be to continue to grieve and mourn the death of Albus Dumbledore, the headmaster of Hogwarts School of Witchcraft and Wizardry.

Output of Llama3-promptCache: Based on the provided search documents, it appears that Harry Potter will be following his friends and classmates out of the Great Hall, as mentioned in the passage: ”Harry looked away from the crowd and followed his friends out of the Great Hall.”

Analysis of Question 1: Upon a meticulous examination of the results of the four models, it can be noticed that the model with block-fine-tuning performs as well as the vanilla-sft model. They successfully deduced Harry’s next action based on the content of the passages. In contrast, for the models without block fine-tuning, namely Llama3-block-w/o-ft and Llama3-promptCache, both have made incorrect inferences: they assume that Harry will remain with his friends. Evidently, in the final conversation between Harry and Ginny, he has already bid his final farewell to Ginny and planned to embark on a solo mission in the future to seek and against Voldemort. Finally, Llama3-promptCache also suffers from a serious context-hallucination problem: the passage ”Harry looked away from the crowd and followed his friends out of the Great Hall.” has never appeared within the context.

User Question 2: Which characters are currently hated by Harry Potter? Among them, who is the one he hates the most?

Output of Llama3-block-ft: Based on the provided search documents, it appears that Harry Potter does not explicitly state who he hates the most. However, it is mentioned that Harry has a strong dislike towards Lord Voldemort, the main antagonist of the series. This is evident in the quote ”I knew this would happen in the end. I knew you wouldn’t be happy unless you were hunting Voldemort.” (from the document titled ”session 10”).

Additionally, Harry also seems to have a strong dislike towards Severus Snape. This is evident in the quote ”His animosity was all for Snape” (from the document titled ”session 2”)

Output of Llama3-vanilla-sft: Based on the provided search documents, Harry Potter currently hates Severus Snape, the Potions Master at Hogwarts.

Output of Llama3-block-w/o-ft: Based on the provided search documents, it appears that Harry Potter does not explicitly state who he hates the most among the characters mentioned. However, it is implied that Harry has a complex relationship with Voldemort, as he mentions that Voldemort is ”the one who had the power to kill Dumbledore” and that he is ”the one who had the power to kill Dumbledore”. This suggests that Harry views Voldemort as a powerful and feared opponent.

However, it is also mentioned that Harry has a special connection to Voldemort, as he says ”I’m the one who had the power to kill Dumbledore” and ”I’m the one who had the power to kill Voldemort”. This implies that Harry feels a sense of responsibility towards Voldemort, as he believes he has the ability to defeat him.

It is worth noting that Harry’s feelings towards Voldemort are complex and multifaceted, and it is not explicitly stated who he hates the most among the characters mentioned.

Output of Llama3-promptCache: Based on the provided search documents, it appears that Harry Potter is not explicitly stated to hate anyone in particular. However, it is mentioned that Harry Potter is not happy with the idea of being used as bait by Voldemort, as he says, ”I can’t say I’m surprised. I knew this would happen. I knew it would happen. I knew it would happen, and I knew I couldn’t stop it.” This suggests that Harry is not happy with the idea of being used as a pawn in Voldemort’s game.

It is also mentioned that Harry’s friends, such as Ron Weasley and Hermione Granger, are not explicitly mentioned as being hated by Harry. However, it is worth noting that Harry’s enemy, Voldemort, is mentioned as being feared and hated by Harry.

Therefore, based on the provided search documents, it can be inferred that Harry Potter hates Voldemort the most.

Analysis of Question 2: The results of question 2 suggest that only the Llama3-block-ft is capable of successfully and comprehensively answering the two sub-questions within question 2. The two models that have not undergone block fine - tuning, namely Llama3-block-w/o-ft and Llama3-promptCache, encounter some knowledge-hallucination issues. Furthermore, Llama3-promptCache also experiences severe degeneration problems.