TokenSelect: Efficient Long-Context Inference and Length Extrapolation for LLMs via Dynamic Token-Level KV Cache Selection

Wei Wu*†
School of Artificial Intelligence and Data Science, University of Science and Technology of China
Hefei, China
urara@mail.ustc.edu.cn

Zhuoshi Pan*†
School of Information Science and Technology, Tsinghua University
Beijing, China
pzs23@mails.tsinghua.edu.cn

Chao Wang
School of Artificial Intelligence and Data Science, University of Science and Technology of China
Hefei, China
chadwang2012@gmail.com

Liyi Chen†
School of Artificial Intelligence and Data Science, University of Science and Technology of China
Hefei, China
liyichencly@gmail.com

Yunchu Bai
School of Management, University of Science and Technology of China
Hefei, China
byc171250@mail.ustc.edu.cn

Kun Fu
Alibaba Cloud Computing
Beijing, China
fk07thu@gmail.com

Zheng Wang‡
Alibaba Cloud Computing
Beijing, China
wz388779@alibaba-inc.com

Hui Xiong‡
Thrust of Artificial Intelligence, The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
Department of Computer Science and Engineering, The Hong Kong University of Science and Technology
Hong Kong SAR, China
xionghui@ust.hk

Abstract

With the development of large language models (LLMs), the ability to handle longer contexts has become a key capability for Web applications such as cross-document understanding and LLM-powered search systems. However, this progress faces two major challenges: performance degradation due to sequence lengths out-of-distribution, and excessively long inference times caused by the quadratic computational complexity of attention. These issues hinder the application of LLMs in long-context scenarios. In this paper, we propose Dynamic Token-Level KV Cache Selection (*TokenSelect*), a model-agnostic, training-free method for efficient and accurate long-context inference. *TokenSelect* builds upon the observation of non-contiguous attention sparsity, using Query-Key dot products to measure per-head KV Cache criticality at token-level. By per-head soft voting mechanism, *TokenSelect* selectively involves a small number of critical KV cache tokens in the attention calculation without sacrificing accuracy. To further accelerate *TokenSelect*, we designed the Selection Cache based on observations of consecutive Query similarity and implemented efficient dot product kernel, significantly reducing the overhead of token selection. A comprehensive evaluation of *TokenSelect* demonstrates up to 23.84× speedup in attention computation and up to 2.28× acceleration in end-to-end latency, while providing superior performance compared to state-of-the-art long-context inference methods.

CCS Concepts

• **Computing methodologies → Natural language generation**.

Keywords

Efficient LLMs Inference, Long-context LLMs, Sparse Attention

1 Introduction

With the rapid development of large language models (LLMs), the number of parameters is no longer the sole factor significantly affecting model performance. The ability to effectively process longer context information has become one of the key metrics for evaluating LLMs' capabilities. The latest Web applications such as cross-document understanding [1], LLM-powered search systems [2], repository-level code completion [3, 4], and complex reasoning [5] have all placed higher demands on the long-context abilities of LLMs. There are two main difficulties in using pre-trained LLMs for long-context inference. On one hand, LLMs are limited by their context length during pre-training (*e.g.* Llama 3 only has 8192 tokens).

*Equal contribution.
†Work done during the internship at Alibaba Cloud Computing.
‡Corresponding authors.

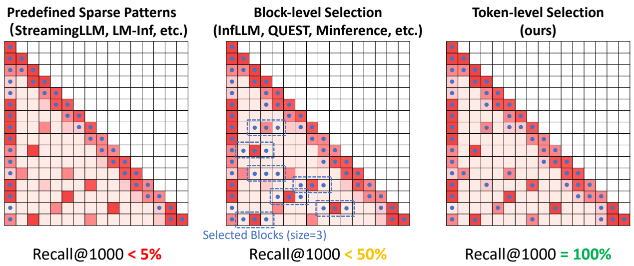


Figure 1: Distribution of tokens participating in attention computation under different sparsity patterns (indicated by blue dots). *TokenSelect* can more accurately select critical tokens for attention computation.

Directly inferencing on longer sequences can lead to severe performance degradation due to reasons including sequence lengths out-of-distribution [6, 7]. On the other hand, even if LLMs possess sufficiently large context lengths, the quadratic computational complexity of attention with respect to sequence length makes the response time for long-context inference unbearable.

Previous works have made numerous attempts to address these difficulties. To extend the context length of LLMs, the current common practice is to perform post-training on long texts [8–10]. However, this approach comes with significant computational costs, particularly in two aspects: the synthesis of high-quality long-text data and the training process on extended sequences. To accelerate long-context inference, many studies focus on the sparsity of attention, attempting to reduce the scale of KV Cache involved in computation. The key to this type of method lies in designing sparse patterns for attention, which can be mainly divided into two categories: one uses predefined sparse patterns [6, 7, 11, 12], while the other estimates the potential importance of KV Cache during the inference process [13–17], attempting to select relevant KV Cache tokens into attention calculations. However, the design of these sparse patterns is often heuristically based on historical criticality or coarse-grained criticality estimation of tokens, making it difficult to ensure that the selected tokens are truly critical, thus resulting in sub-optimal performance, as shown in *Fig.* 1.

In this paper, we further observe the non-contiguous sparsity of attention, revealing the importance of designing more fine-grained dynamic sparse patterns. To this end, we propose *TokenSelect*, a model-agnostic and training-free approach that utilizes token-level selective sparse attention for efficient long-context inference and length extrapolation. Specifically, for each Query, *TokenSelect* dynamically calculates token-level per-head criticality for the past KV Cache and selects the k most critical tokens through our head soft vote mechanism, involving them in the attention calculation. This reduces the scale of attention calculation to a constant length familiar to the model, while maintaining almost all of the long-context information, thereby simultaneously addressing the two main difficulties for long-context inference. To reduce the overhead of token selection, *TokenSelect* manages the KV Cache in token-level pages [18] and design efficient kernel for token selection based on Paged KV Cache management through Triton [19]. Furthermore, based on our observation of high similarity between consecutive queries, we have designed the Selection Cache, which allows consecutive similar queries to share token selection results, thereby reducing the selection frequency while ensuring its effectiveness.

We evaluate the performance and efficiency of *TokenSelect* on three representative long-context benchmarks [1, 20, 21] using three open-source LLMs [9, 22, 23]. The experimental results demonstrate that our *TokenSelect* can achieve up to 23.84× speedup in attention computation compared to FlashInfer [24], and up to 2.28× acceleration in end-to-end inference latency compared to state-of-the-art long-context inference method [15]. Simultaneously, it provides superior performance on three long-text benchmarks. In summary, we make the following contributions:

- An observation on the non-contiguous sparsity of attention that highlights the importance of token-level selection.
- *TokenSelect*, a model-agnostic and training-free method that achieves accurate and efficient long-context inference and length extrapolation, which is compatible with mainstream LLM serving systems and ready for Web applications.
- A comprehensive evaluation of *TokenSelect*, demonstrating up to 23.84× speedup in attention computation and up to 2.28× acceleration in end-to-end latency while exhibiting superior performance.

2 Related Works

Long-context LLMs. Due to computational complexity constraints, current LLMs based on Transformers often utilize limited context lengths during pre-training [9, 10, 22, 23, 25, 26]. To extend the long-context capabilities of LLMs, current methods can be broadly categorized into three approaches [27–29]: 1) Modifying positional encodings: A widely adopted method is positional interpolation [30]. Chen et al. first proposed linear scaling of RoPE [31] to map longer positional ranges within the original training window. Subsequent works [32, 33] further improved this method using Neural Tangent Kernel (NTK) theory [34], achieving longer context windows while maintaining model performance. Methods like YaRN [35] and Giraffe [36] optimize interpolation effects by adjusting frequency components or introducing temperature parameters. 2) Long-context post-training: This approach extends the model's context length through additional training steps on longer documents after pre-training [37, 38]. It has been widely adopted by leading LLMs [8–10] with the support of sequence parallelism techniques [39–41]. 3) Incorporating additional memory modules: Notable examples include Transformer-XL [42], Compressive Transformer [43], RMT [44] and Infini-attention [45]. Although these methods have expanded the context length of LLMs, long-context inference still faces the challenge of high computational costs.

Efficient Long-context Inference. In state-of-the-art LLMs serving systems [18, 46–48], technologies such as Flash Attention [49, 50] and Paged Attention [46] have greatly optimized LLMs inference efficiency by improving GPU I/O bottlenecks. However, in long-context inference scenarios, the quadratic computational complexity of attention with respect to sequence length poses new challenges for LLMs inference. Numerous studies focus on the sparsity of attention, selecting partial KV Cache for attention calculations to improve long-context inference efficiency. Sliding window [11, 12] is one of the most widely used sparse patterns, reducing complexity to linear by executing attention computations within localized windows. Recent works like StreamingLLM [6] and LM-infinite [7] retain the initial tokens of the sequence in addition to sliding windows, effectively maintaining LLMs' performance when processing long sequences. While these approaches are simple to implement, they cannot retain information from long contexts. Another approach focuses on dynamic KV Cache selection during inference. Methods like H2O [13], TOVA [14], FastGen [51], Scissorhands [52], and SnapKV [53] evaluate token criticality based on historical attention scores, selecting tokens within a limited budget. However, these methods permanently discard parts of the KV Cache, causing information loss from long contexts. To address this, InfLLM [15] introduces Block Memory Units for KV Cache management, retrieving information from long contexts and offloading less-used blocks to CPU. Similarly, QUEST [16] proposes query-aware sparsity at page granularity, while MInference [17] optimizes long-context inference using three sparse patterns. Apart from considering all attention heads, some other works [54–56] attempt to focus on only a subset of attention heads. Beyond selection, some other research focuses on KV Cache quantization [57–60] and merging [61–64]. However, existing methods struggle to be applied in real-world Web applications, both in terms of accuracy and efficiency.

3 Preliminaries

In this section, we first introduce the inference process of LLMs, and then define the Selective Sparse Attention Problem.

3.1 LLMs Inference

Nowadays, mainstream LLMs are primarily based on the Decoder-only Transformer architecture, consisting sequentially of a word embedding layer, a series of transformer layers, and a token prediction head. Each transformer layer includes a multi-head attention (MHA) module and a feed-forward networks (FFN) module. The inference process of LLMs can be divided into two stages: the Prefill Stage and the Decode Stage.

The Prefill Stage is the preparatory phase of the inference process. In this stage, the user's input is processed layer by layer through a single forward pass of LLMs, generating KV Cache for each layer. The generation of KV Cache is completed by the MHA module. Assuming $\mathbf{X}_{\text{prefill}} \in \mathbb{R}^{n_{\text{in}} \times d}$ is the input of a transformer layer, where n_{in} is the number of tokens in user's input sequence and d is the hidden size. The computation of MHA in the Prefill Stage is as follows (simplified to single-head mode):

$$\left[\mathbf{Q}_{\text{prefill}}, \mathbf{K}_{\text{prefill}}, \mathbf{V}_{\text{prefill}}\right] = \mathbf{X}_{\text{prefill}} \cdot \left[\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v\right], \tag{1}$$

$$\mathbf{O}_{\text{prefill}} = \text{softmax}\left(\frac{\mathbf{Q}_{\text{prefill}} \cdot \mathbf{K}_{\text{prefill}}^{\top}}{\sqrt{d}}\right) \cdot \mathbf{V}_{\text{prefill}}, \tag{2}$$

where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v$ are linear projections, $[\cdot]$ represents tensor concatenation operation, and *Eq.*(2) is also known as Scaled Dot-Product Attention (SDPA). After these computation, $\mathbf{K}_{\text{prefill}}$ and $\mathbf{V}_{\text{prefill}}$ are stored as the KV Cache for current layer $\mathbf{K}_{\text{cache}}$ and $\mathbf{V}_{\text{cache}}$, and $\mathbf{O}_{\text{prefill}}$ is used for subsequent calculations.

The Decode Stage is the phase where LLMs actually generate the response. In the Decode Stage, LLMs load the KV Cache and generate n_{out} output tokens autoregressively through n_{out} forward passes. Assuming $\mathbf{X}_{\text{decode}} \in \mathbb{R}^{1 \times d}$ is the input of a transformer layer in a forward pass, the computation of MHA in the Decode Stage is as follows (The calculation of $\mathbf{Q}_{\text{prefill}}$ and $\mathbf{O}_{\text{prefill}}$ is consistent with that in the Prefill Stage):

$$\begin{aligned}\mathbf{K}_{\text{decode}} &= [\mathbf{K}_{\text{cache}},\ \mathbf{X}_{\text{decode}} \cdot \mathbf{W}_k],\ \mathbf{K}_{\text{cache}} \leftarrow \mathbf{K}_{\text{decode}},\\ \mathbf{V}_{\text{decode}} &= [\mathbf{V}_{\text{cache}},\ \mathbf{X}_{\text{decode}} \cdot \mathbf{W}_v],\ \mathbf{V}_{\text{cache}} \leftarrow \mathbf{V}_{\text{decode}},\end{aligned} \tag{3}$$

where $\mathbf{K}_{\text{decode}}, \mathbf{V}_{\text{decode}}$ are composed of the KV Cache and the KV corresponding to the current input, which are then used to update the KV Cache of the current layer for use in the next forward pass.

LLMs inference, unlike training, is memory-bound, necessitating frequent GPU I/O operations between HBM and SRAM while underutilizing processing units. This bottleneck is particularly evident in SDPA computation. Optimizing for I/O is crucial for enhancing LLMs inference efficiency, especially in long-context scenarios.

3.2 Selective Sparse Attention

As discussed in the *Sec.* 1, the high attention sparsity in LLMs suggests sparse attention as a promising solution for long-context inference challenges. Sparse attention can keep the number of tokens participating in attention computations at a constant scale, rather than increasing with sequence length. Given that predefined sparse patterns are detrimental to performance, we aim to dynamically select crucial tokens for attention computation at each step during the inference process. Therefore, we formalize this problem according to the following definition.

Definition 1 (Selective Sparse Attention Problem, informal). *For current input of length C ($C = 1$ in the Decode Stage) and KV Cache of length N, assuming there are H attention heads with a head size of d_h, let $\mathbf{O}$ be the output of the SDPA:*

$$\mathbf{O} = \left[softmax\left(\frac{\mathbf{Q}^h \cdot \left[\mathbf{K}^h_{cache},\ \mathbf{K}^h_{current}\right]^{\top}}{\sqrt{d}}\right) \cdot [\mathbf{V}^h_{cache},\ \mathbf{V}^h_{current}] \right]_{h \in \{1, \cdots, H\}}, \tag{4}$$

where $\mathbf{Q}^h, \mathbf{K}^h_{current}, \mathbf{V}^h_{current} \in \mathbb{R}^{C \times d_h}$ is the Query, Key, Value matrices of current input for head h and $\mathbf{K}^h_{cache}, \mathbf{V}^h_{cache} \in \mathbb{R}^{N \times d_h}$ represent the KV Cache. Let $\hat{\mathbf{O}}$ be the output of the Selective Sparse Attention:

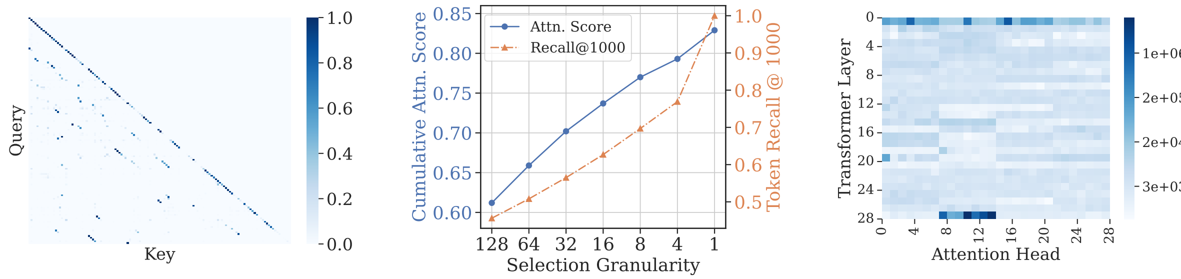
$$\hat{\mathbf{O}} = \left[softmax\left(\frac{\mathbf{Q}^h \cdot \left[\mathbf{K}^h_{select},\ \mathbf{K}^h_{current}\right]^{\top}}{\sqrt{d}}\right) \cdot [\mathbf{V}^h_{select},\ \mathbf{V}^h_{current}] \right]_{h \in \{1, \cdots, H\}}, \tag{5}$$

where $\mathbf{K}^h_{select}, \mathbf{V}^h_{select} \in \mathbb{R}^{k \times d_h}$ are k selected KV Cache ($k \ll N$). The selection of $\mathbf{K}_{select}, \mathbf{V}_{select}$ is performed by selection function $\mathcal{S}$:

$$\begin{aligned}\mathcal{S}\left(\mathbf{Q},\ \mathbf{K}_{cache}\right) &= \mathcal{I},\ \ where\ \mathcal{I} \in \mathcal{P}(\{1, \cdots, N\}),\\ \mathbf{K}_{select} = [(\mathbf{K}_{cache})_i]_{i \in \mathcal{I}},&\ \mathbf{V}_{select} = [(\mathbf{V}_{cache})_i]_{i \in \mathcal{I}},\end{aligned} \tag{6}$$

where $\mathcal{I}$ is the set of selected indices. The objective is to find an appropriate selection function $\mathcal{S}$ that minimizes the difference between the outputs of the SDPA and the selective sparse attention:

$$\min_{\mathcal{S}} \left\| \mathbf{O} - \hat{\mathbf{O}} \right\|_2^2. \tag{7}$$

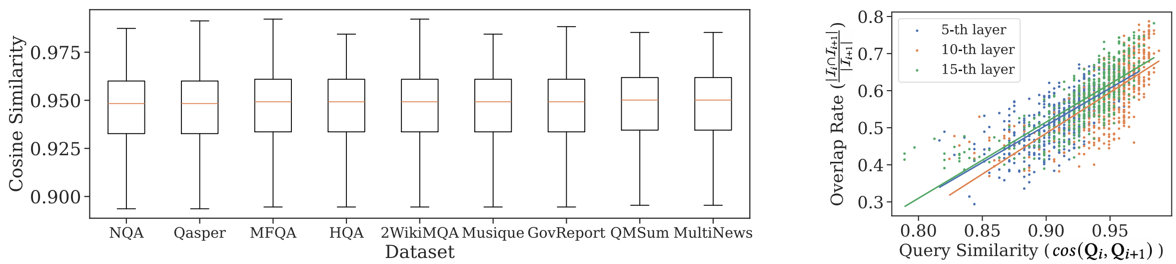


(a) Attention is sparse in token-level.

(b) Block-level selection is sub-optimal.

(c) Attention logits is head-distinctive.

Figure 2: Motivations for token-level selection. (a) Visualization of attention scores sparsity. (b) Attention scores and critical token recalled by 1K token budget. (c) The L_1 norm of attention logits in each attention head. Visualizations are based on `Qwen-2-7B-Instruct` on the GovReport dataset of *LongBench*.



(a) Consecutive query is similar and the similarity distribution is consistent across datasets.

(b) Selection overlaps when query is similar.

Figure 3: Observations on similarity of consecutive queries. (a) Cosine similarity distribution between consecutive queries. (b) The token selection overlap rate ($\frac{|\mathcal{I}_i \cap \mathcal{I}_{i+1}|}{|\mathcal{I}_{i+1}|}$) with respect to consecutive query similarity.

Existing works on long-context inference [6, 7, 11–17, 53] can be categorized under the Selective Sparse Attention Problem, with variations primarily in the design of the selection function $\mathcal{S}$. [6, 11, 12] have developed input-independent selection functions $\mathcal{S}()$, while [13, 14, 53] propose query-independent functions $\mathcal{S}(\mathbf{K}_{\text{cache}})$ for improved performance. Current state-of-the-art methods [15–17] utilize query-aware selection functions $\mathcal{S}(\mathbf{Q}, \mathbf{K}_{\text{cache}})$. However, these approaches typically operate at a block-level, which limits their effectiveness and overall performance.

4 Motivations and Observations

Attention is Sparse, Non-contiguous and Head-Distinctive. Previous works [6, 7, 11–17, 53] have demonstrated the sparsity of attention scores in LLMs, particularly when processing long texts. Recent approaches [15–17] partition the KV Cache into non-overlapping blocks, estimating block criticality for sparse attention calculations. These methods assume that tokens with higher attention scores tend to be contiguous. However, our further observations reveal that this assumption does not always hold true in practice. As illustrated in *Fig.* 2a, attention scores are sparsely distributed at the token-level, with critical tokens not necessarily contiguous. This non-contiguity leads to significant omissions in block-level token selection. *Fig.* 2b demonstrates that finer selection granularity improves recall of critical tokens, motivating us to perform token-level selection. For token-level selection, an intuitive approach would be to directly select the top-k tokens with the highest attention logits. However, observation in *Fig.* 2c reveals considerable disparity in the L_1 norm of attention logits across attention heads. As a result, the selection result tends to be dominated by a few heads with disproportionately large attention logits, driving us to design a more robust selection function that maintains the independence of heads.

Consecutive Queries are similar. As sparsity of attention is dynamic [13, 15–17], token selection should be performed for every Query, which inevitably increases the computational overhead of selective sparse attention. Fortunately, we observe that consecutive Queries exhibit high similarity, as shown in *Fig.* 3a. Intuitively, when two consecutive Queries are highly similar, their dot products with the Keys will also be similar, leading to substantial overlap in the token selection results. Due to space constraints, we provide an informal lemma about this below. The formal version and corresponding proof can be found in the Appendix A.

Lemma 1 (Informal). *Consider Queries $\mathbf{Q}_1, \mathbf{Q}_2 \in \mathbb{R}^{1\times d}$ that are consecutive and a Key set $\{\mathbf{K}_i\}_{i=1}^N$. Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be the sets of indices of the top-k Keys selected by dot product for $\mathbf{Q}_1$ and $\mathbf{Q}_2$ respectively. If $cos(\mathbf{Q}_1, \mathbf{Q}_2) > \epsilon$, where ϵ is a threshold, then $\mathcal{I}_1 = \mathcal{I}_2$.*

Fig. 3b illustrates this lemma experimentally. It can be seen that the overlap rate of token selection tends to increase with query similarity. This key insight motivates us to reuse selection results for similar queries, improving computational efficiency. Moreover, the similarity distribution of consecutive Queries remains consistent

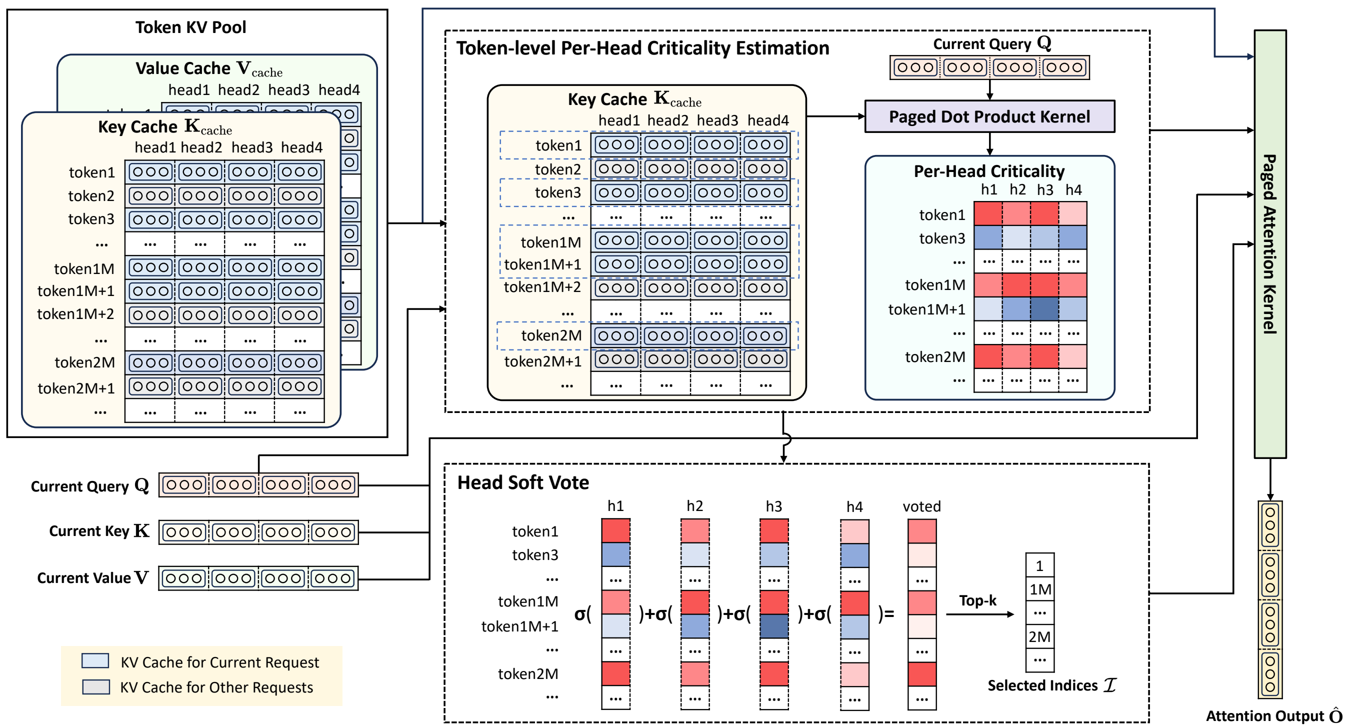


Figure 4: The illustration of *TokenSelect*, which involves calculating per-head criticality using the Paged Dot Product Kernel, performing head soft vote to get selected indices, and executing selective sparse attention via the Paged Attention Kernel.

across different tasks, as demonstrated in *Fig.* 3a, allowing us to apply a global similarity threshold across all scenarios.

5 Designs of *TokenSelect*

In this section, we will introduce the design details of *TokenSelect*, primarily encompassing the Selection Function, the Selection Cache, and efficient implementation of *TokenSelect*. The overall workflow of *TokenSelect* is illustrated in *Fig.* 4.

5.1 Selection Function

The simplest selection function is to determine the criticality of the tokens through the dot product of $\mathbf{Q}$ and $\mathbf{K}_{\text{cache}}$, then select the top-$k$ critical ones as $\mathbf{K}_{\text{select}}, \mathbf{V}_{\text{select}}$. The selected indices $\mathcal{I}$ are calculated as follow:

$$\mathcal{I}_{\text{topk}} = \text{TopK}\left(\mathbf{Q} \cdot {\mathbf{K}^h_{\text{cache}}}^\top\right). \tag{8}$$

However, as discussed in *Sec.* 4, this approach is prone to inaccuracies due to disparities in norm of attention logits between heads. To maintain independence between heads, a better approach is to have each head select the top-k most critical tokens, and then determine the final selection through voting among the heads:

$$\mathcal{I}_{\text{head-vote}} = \text{TopK}\left(\sum_{h=1}^{H} \mathbb{I}\left(i \in \text{TopK}\left(\mathbf{Q}^h \cdot {\mathbf{K}^h_{\text{cache}}}^\top\right)\right)\right), \tag{9}$$

where $\mathbb{I}$ is the indicator function. Unfortunately, despite better performance, this method relies on `scatter_add` and multiple `topk` operations, resulting in low efficiency on GPUs. Additionally, the 0/1 voting ignores the relative importance of tokens for each head.

Therefore, we propose a head soft vote approach that offers better performance and efficiency. Specifically, we first calculate the per-head criticality, then normalize through softmax, and sum the results for all heads. The formalization is as follows:

$$\mathcal{I}_{\text{head-soft-vote}} = \text{TopK}\left(\sum_{h=1}^{H} \text{softmax}\left(\mathbf{Q}^h \cdot {\mathbf{K}^h_{\text{cache}}}^\top\right)\right). \tag{10}$$

5.2 Optimizing Selection Frequency

Although the aforementioned selection function can reduce the complexity of attention from $O(N^2)$ to $O(k^2)$ ($k \ll N$), while maintaining performance, the execution time of the selection function itself still affects the latency of LLMs inference. To further accelerate long-context inference, based on our observations of the similarity of consecutive queries, we design optimization strategies for both the Prefill Stage and the Decode Stage to reduce the selection frequency while ensuring its effectiveness.

In the Prefill Stage, $\mathbf{Q}_{\text{prefill}} \in \mathbb{R}^{n_{\text{in}} \times d}$ is inputed. In long-context scenarios, the number of tokens in the user's input sequence n_{in} may reach up to 1M, making it impractical to perform selection for each Query token. Considering the similarity of consecutive Queries, we use chunk-wise token selection, inputting $\frac{1}{c}\sum_{i=1}^{c}(\mathbf{Q}_C)_i$ into the selection function, where $\mathbf{Q}_C \in \mathbb{R}^{c \times d}$ is the Query chunk and c is the chunk size. This method helps maintain the compute-intensive nature of the Prefill Stage, preventing it from becoming memory bound.

In the Decode Stage, due to the auto-regressive characteristic of LLMs, we need to frequently perform selection for $\mathbf{Q}_{\text{decode}} \in \mathbb{R}^{1 \times d}$,

Algorithm 1 Selection Cache Algorithm for the Decode Stage

Require: Q: current Query, k: number of selected tokens,
C_Q: Query cache, $\mathrm{C}_{\mathcal{I}}$: selection cache,
$\mathcal{S}$: selection function (*Eq.*(10)), θ: similarity threshold,
f: first query flags (default True)
Ensure: $\mathcal{I}$: selected indices

```
if f or cos(Q, C_Q) < θ then
    I ← S(Q, k)
    C_I ← I
    C_Q ← Q
    f ← False
else
    I ← C_I
end if
return I
```

and this process cannot be executed chunk-wise like in the Prefill Stage. To reduce the frequency of token selection in the Decode Stage, as shown in Algorithm 1, we propose the Selection Cache. Consecutive similar Queries will hit the cache, thereby directly loading the cached selection results for the previous Query. The Selection Cache allows us to reduce decode latency while maintaining almost the same performance.

5.3 Efficient Implementation

To ensure that our proposed *TokenSelect* can be used for real-world Web applications, efficient implementation is crucial. We first analyze the computation time breakdown of representative block-level selective sparse attention method, InfLLM [15]. From (1)(2)(3) in *Fig.* 5, we can observe that although selective sparse attention can significantly reduce the complexity of attention calculations, the actual computation time is still highly dependent on the implementation. The incompatibility with efficient attention implementations such as Flash Attention has resulted in methods requiring historical attention scores [13–15, 53] being difficult to be applied in real-world Web applications. Through the analysis of InfLLM's Flash Attention-compatible version, we make several discoveries. The initial motivation for estimating token criticality at the block-level is to reduce the overhead of selection function (mainly considering dot product calculation). However, we find that dot product is not the primary performance bottleneck. Instead, a significant portion of the overhead comes from indexing the KV Cache using selected indices and making them contiguous in GPU memory, which frequently occurs during the updating of KV blocks and the concatenation of selected KV Cache. The extensive I/O required for this operation further exacerbates the memory-bound in LLMs inference. Based on this, we propose that Paged Attention is a more suitable implementation for selective sparse attention. Using Paged KV Cache management (with page size=1 for *TokenSelect*), we can reduce the I/O volume for selection results from the scale of all selected KV Caches $O(2kd)$ to the scale of their indices $O(k)$. However, by observing (4) in *Fig.* 5, we find that we encounter another bottleneck under Paged KV Cache management. Since logically contiguous KV Cache is not entirely contiguous in GPU memory, it also needs to be made contiguous before performing computational operations. To address this issue, we draw inspiration from the concept of Paged Attention and implement a Paged Dot Product

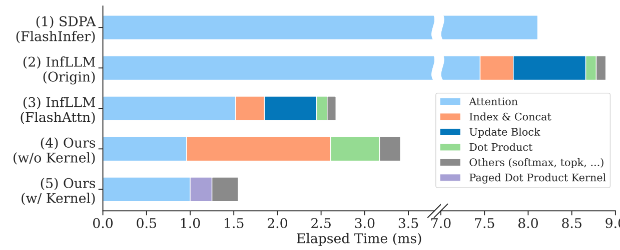


Figure 5: Computation time breakdown for single chunk prefill step under different attention implementations (chunk size: 512, KV Cache length: 128K, attended tokens: 4K).

Kernel using Triton [19], which significantly improves the overall efficiency of *TokenSelect*.

6 Experiments

In this section, we first introduce the experimental setup of this paper, and then reveal the performance and efficiency of our *TokenSelect* in long-context inference through experiments.

6.1 Experimental Settings

Datasets. To evaluate *TokenSelect*'s performance on long-context inference, we use the following datasets: (1) InfiniteBench [20]: The mainstream long-context benchmark consisting of multi-tasks. The average length of it exceeds 200K tokens. (2) RULER [21]: A challenging long-context benchmark containing 13 different tasks, with subsets of varying lengths up to 128K tokens. (3) LongBench [1]: Another mainstream long-context benchmark comprising 6 types of tasks. The 95% percentile for its lengths is 31K tokens. For each dataset, we use its recommended metrics, which are presented in the Appendix B.

Baselines. To demonstrate the state-of-the-art (SOTA) performance of *TokenSelect*, we include the following methods for comparison: (1) **Original** models: We select three mainstream open-source LLMs - `Qwen2-7B-Instruct` [9], `Llama-3-8B-Instruct` [22], and `Yi-1.5-6B-Chat` [23] - utilizing their original context lengths without any modifications. (2) **NTK**-Aware Scaled RoPE: A nonlinear RoPE interpolation method. (3) **SelfExtend**: A RoPE interpolation method that reuses the position ids across neighboring tokens. (4) **StreamingLLM**: The SOTA method for long-context inference with predefined sparse patterns. (5) **InfLLM**: The SOTA method for long-context inference and length extrapolation using a block-level selective sparse attention method. (6) **MInference**: The SOTA method for long-context prefilling acceleration, utilizing three sparse patterns including block-level sparse attention. It's worth noting that since MInference doesn't support length extrapolation, we use an alternative evaluation method, applying it to `Llama-3-8B-Instruct-262k` (Llama3 after long-text post-training). Additionally, we do not include another SOTA method, QUEST [16], as it does not support Grouped Query Attention (GQA).

Implementation details. In all experiments in this paper, we employ greedy decoding to ensure the reliability of the results. For our *TokenSelect*, we implement it on SGLang [18], which is a fast serving framework based on Flasherinfer [24]. We implement our method using PyTorch [65] and Triton [19]. We follow the baseline

Table 1: Comparison of different methods with different origin models on InfiniteBench.

Methods	En.Sum	En.QA	En.MC	En.Dia	Code.D	Math.F	R.PK	R.Num	R.KV	Avg.
Qwen2-7B	23.80	14.92	54.59	8.50	28.17	19.71	28.81	28.64	19.00	25.13
NTK	18.73	15.34	41.28	**7.50**	24.87	27.71	99.15	97.46	59.80	43.54
SelfExtend	3.76	4.44	20.09	5.00	8.12	2.29	0.00	0.00	0.00	4.86
StreamingLLM	19.60	13.61	48.03	3.50	27.92	19.43	5.08	5.08	2.40	16.07
InfLLM	19.65	15.71	46.29	**7.50**	27.41	24.00	70.34	72.20	5.40	32.06
TokenSelect	**22.62**	**18.86**	**48.47**	**7.50**	**30.20**	**32.57**	**100.00**	**100.00**	**86.60**	**49.65**
Llama-3-8B	24.70	15.50	44.10	7.50	27.92	21.70	8.50	7.80	6.20	18.21
NTK	6.40	0.40	0.00	0.00	0.50	2.60	0.00	0.00	0.00	1.10
SelfExtend	14.70	8.60	19.70	0.00	0.00	22.60	**100.00**	**100.00**	0.20	29.53
StreamingLLM	20.40	14.30	40.60	5.00	**28.43**	21.40	8.50	8.30	0.40	16.37
InfLLM	24.30	**19.50**	43.70	10.50	27.41	23.70	**100.00**	99.00	5.00	39.23
TokenSelect	**26.99**	19.39	**45.85**	**14.50**	27.41	**28.29**	**100.00**	97.29	**40.00**	**44.41**
Yi-1.5-6B	18.78	10.48	39.74	5.00	29.95	16.00	5.08	5.08	0.00	14.45
NTK	4.66	0.58	0.87	0.00	0.00	1.43	0.00	0.00	0.00	0.83
SelfExtend	5.62	1.07	1.31	0.00	0.00	1.14	0.00	0.00	0.00	1.01
StreamingLLM	15.35	9.26	35.81	5.00	27.41	14.29	5.08	4.92	0.00	13.01
InfLLM	16.98	8.93	34.06	3.00	27.41	16.86	**100.00**	96.61	0.00	33.76
TokenSelect	**21.13**	**12.32**	**40.61**	**5.50**	**30.71**	**20.86**	**100.00**	**99.83**	0.00	**36.77**

Table 2: Comparison of different methods with different origin models on RULER.

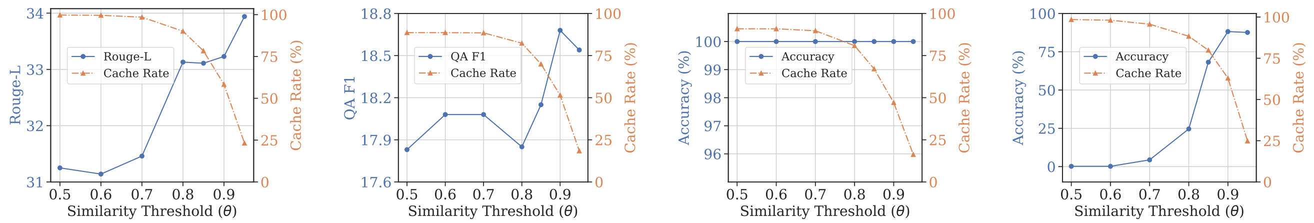
Methods	4K	8K	16K	32K	64K	128K	Avg.
Qwen2-7B	90.74	84.03	80.87	79.44	74.37	64.13	78.93
StreamingLLM	94.41	54.59	33.54	22.40	15.38	10.88	38.53
InfLLM (2K+512)	52.85	36.09	29.36	23.52	18.81	18.29	29.82
InfLLM (4K+4K)	55.22	52.10	40.53	29.77	21.56	18.64	36.30
Ours (2K+512)	94.11	81.81	68.68	60.62	51.81	42.75	66.63
Ours (4K+4K)	**94.42**	**90.22**	**82.06**	**70.40**	**59.66**	**54.28**	**75.17**
Llama-3-8B	93.79	90.23	0.09	0.00	0.00	0.00	30.69
StreamingLLM	93.68	54.48	33.77	20.35	14.88	11.47	38.11
InfLLM (2K+512)	79.79	52.43	40.12	33.60	25.68	23.39	42.50
InfLLM (4K+4K)	93.79	86.11	64.33	45.39	33.13	27.81	58.43
Ours (2K+512)	93.73	82.92	**71.92**	**65.38**	**59.35**	33.39	**67.78**
Ours (4K+4K)	**93.88**	**90.29**	70.13	57.72	48.36	**39.38**	66.63
Yi-1.5-6B	73.12	9.09	0.37	0.01	0.00	0.01	13.77
StreamingLLM	72.10	33.03	21.69	15.39	12.58	12.61	27.90
InfLLM (2K+512)	59.66	36.77	27.41	24.49	21.49	21.17	31.83
InfLLM (4K+4K)	74.81	52.57	27.65	22.83	20.19	19.48	36.26
Ours (2K+512)	**75.93**	**59.55**	**49.69**	**42.36**	**34.68**	**31.36**	**48.93**

approach, including 128 initial tokens and n_{local} most recent tokens in the attention computation in addition to the k selected tokens. For NTK and SelfExtend, we extend the model's context length to 128K. For StreamLLM, we set $n_{\text{local}} = 4\text{K}$. For InfLLM, we set $k = 4\text{K}, n_{\text{local}} = 4\text{K}$. For our *TokenSelect*, we set $k = 2\text{K}, n_{\text{local}} = 512$ to demonstrate our token-level KV Cache selection allows us to achieve better performance with a smaller token budget. Due to the need to demonstrate the method under different n_{local} and k, we denote the specific token budgets in the form of $k + n_{\text{local}}$ if they differ from the aforementioned settings. We use NVIDIA A100 to conduct all experiments. When inferencing sequences over 1M tokens, we additionally employee tensor parallelism, which is transparent to our *TokenSelect*.

6.2 Performance Comparisons

InfiniteBench. As shown in Table 1, our *TokenSelect* achieves significantly superior overall performance on InfiniteBench compared to all baseline methods, even though *TokenSelect* uses the smallest token budget (<3K). The fact that it significantly outperforms the original models demonstrates *TokenSelect*'s strong length extrapolation capability. We analyze that this is due to our adoption of a fine-grained KV Cache selection strategy, while considering the equal contribution of each head to selection, which ensures that we can select most critical tokens. Observing the performance of other methods, we find that RoPE interpolation methods (NTK, SelfExtend) generally perform poorly unless used on specially trained models such as `Qwen2-7B-Instruct`. The better performance of `Qwen2-7B-Instruct` on the original model can also be attributed to this. The sparse attention method StreamingLLM, based on fixed sparse patterns, can guarantee some of the model's capabilities, but due to discarding a large amount of long-context information, it performs poorly on retrieval-related tasks (R.PK, R.Num, R.KV). The block-level selection method InfLLM can retain more long-context information compared to StreamingLLM. However, due to its sub-optimal block-level selection, it results in lower performance on most tasks compared to *TokenSelect*, even though we set a larger token budget for InfLLM. It is worth noting that `Yi-1.5-6B` does not perform normally on the R.KV task, as it is unable to correctly recite strings like the Universally Unique Identifier.

RULER. To further demonstrate the long-context capability of *TokenSelect*, we conduct evaluation on the more challenging long-context benchmark RULER. Considering the increased difficulty of RULER and its substantial computational requirements, we include only comparable baseline methods. As shown in Table 2, our *TokenSelect* maintains significantly superior overall performance compared to other long-context inference methods. For all models, *TokenSelect* achieves length extrapolation while preserving the model's original capabilities, benefiting from our efficient utilization of the model's limited context length. Notably, due to the



(a) LongBench-GovReport.
(b) InfiniteBench-En.QA.
(c) InfiniteBench-R.PK.
(d) InfiniteBench-R.KV.

Figure 6: Performance and Cache Rate with different similarity threshold θ of the Selection Cache on `Qwen2-7B-Instruct`.

Table 3: Comparison of different methods on post-trained models on InfiniteBench and LongBench.

Methods	InfiniteBench					LongBench
	En.Sum	En.QA	Code.D	Math.F	R.KV	Avg.
LLaMA-3-8B-262K	20.2	12.4	22.1	26.6	14.4	33.9
+ MInference	20.5	12.9	22.3	**33.1**	12.8	38.4
Ours (w/ *LLaMA-8K*)	**26.9**	**19.3**	**27.4**	28.2	**40.0**	**44.0**

Table 4: Ablation study of the Selection Function $\mathcal{S}$ on InfiniteBench using `Qwen2-7B-Instruct`.

$\mathcal{S}$	En.QA	En.MC	Code.D	Math.F	R.Num	R.KV
$\mathcal{I}_{\text{topk}}$	15.15	45.85	28.43	31.14	98.47	16.60
$\mathcal{I}_{\text{head-vote}}$	17.01	45.85	28.68	30.86	100.00	22.40
$\mathcal{I}_{\text{head-soft-vote}}$	**18.86**	**48.47**	**30.20**	**32.57**	**100.00**	**86.60**

Table 5: Performance vs. Number of selected tokens k on InfiniteBench using `Qwen2-7B-Instruct`.

k	En.Sum	En.QA	En.Mc	Math.F	R.Num	R.KV
128	21.23	10.46	41.48	18.00	100.00	13.40
256	22.01	11.66	41.92	19.71	100.00	20.00
512	21.60	13.31	40.17	21.71	100.00	45.60
1K	21.35	15.13	44.10	24.57	100.00	73.00
2K	22.62	18.86	48.47	**32.57**	100.00	86.60
4K	24.09	21.11	51.53	21.71	100.00	**88.00**
8K	25.32	22.93	58.52	23.71	100.00	85.40
16K	**26.54**	**23.04**	**62.88**	28.16	**100.00**	72.00

constraints of model's context length, *TokenSelect* experiences performance degradation with larger token budgets (4K+4K) on Llama and Yi. However, its performance with smaller token budgets still significantly surpasses other baseline methods.

LongBench. Due to space constraints, the results of LongBench are presented in the Appendix C. Although its relatively shorter text length makes it less suitable for evaluating state-of-the-art long-context inference methods, our *TokenSelect* still demonstrates superior overall performance compared to most baseline methods.

Comparing to methods based-on post-trained model. In Table 3, we present the performance of the post-trained model and long-context inference method [17] based on it. It shows that even compared to length extrapolation methods requiring additional training, the training-free *TokenSelect* still exhibits superior performance on most tasks. Although Minference can improve the performance of the original model, it fails to reverse the negative impact of long-text post-training on shorter text tasks (LongBench).

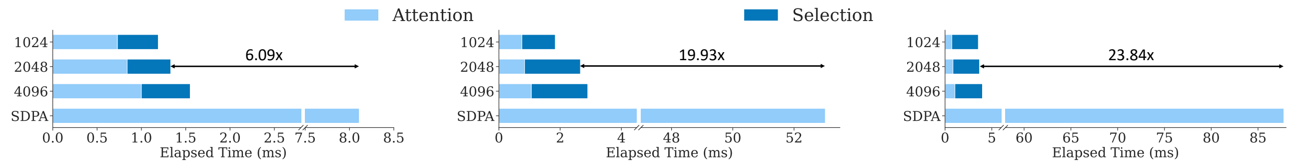
6.3 Ablation Studies

In ablation studies, we primarily analyze the impact of different Selection Functions $\mathcal{S}$ on performance. To compare the performance of different Selection Functions $\mathcal{S}$ under low token budgets (*i.e.*, token efficiency), we maintain the 2K+512 configuration. From Table 4, we can observe that our proposed head soft vote mechanism performs significantly better across all tasks. This indicates that using the head soft vote mechanism to balance each head's contribution to token selection results can help us avoid the domination of selection by few heads with large attention logits.

6.4 Hyper-parameter Analysis

Number of selected tokens k. As shown in Table 5, we fixed n_{local} to a relatively small value (512) to compare the performance when selecting different numbers of tokens. First, we observe that even selecting a very small number of tokens (*e.g.*, 128, 256), our *TokenSelect* still demonstrates very comparable performance. Then, as k increases, the effectiveness of *TokenSelect* further improves, indicating that more moderately critical tokens also contribute to the retention of long-context information. Finally, we find that when k is set to larger values (*e.g.*, 16K), our *TokenSelect* shows significant improvements in most tasks, further advancing the performance landscape of long-context inference methods.

Similarity threshold of the Selection Cache θ. *Fig.* 6 shows that the Selection Cache hit rate increases significantly as the similarity threshold θ decreases, converging around $\theta = 0.5$. This suggests potential for further acceleration of *TokenSelect*'s Decode Stage by reducing θ. Performance sensitivity to θ varies across tasks. While most tasks exhibit slight performance degradation with decreasing θ, and R.PK in InfiniteBench shows no degradation, more challenging retrieval tasks like R.KV demonstrate significant performance deterioration. This indicates higher dynamicity requirements for token selection in these tasks.



(a) KV Cache length: 128K. (b) KV Cache length: 512K. (c) KV Cache length: 1M.

Figure 7: Computation time for single chunk prefill step on different KV Cache lengths using `Qwen2-7B-Instruct`. The vertical axis represents the number of attended tokens, where SDPA denotes full attention by Flashinfer (chunk size: 512).

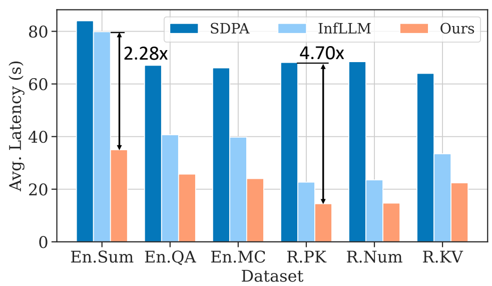


Figure 8: End to end latency per sample with different methods on InfiniteBench using `Qwen2-7B-Instruct`.

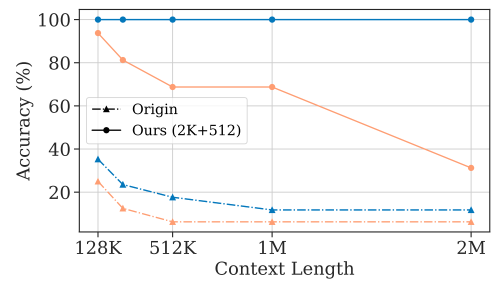


Figure 9: Performance comparison on extended R.PK and R.KV with different text length using `Qwen2-7B-Instruct`.

6.5 Efficiency Comparisons

Efficiency of selective sparse attention. *Fig.* 7 demonstrates the significant acceleration of attention computation achieved by *TokenSelect* during long-context inference. With a KV Cache length of 1M, *TokenSelect* can provide up to 23.84× speedup compared to FlashInfer, which is the inference kernel library we based on. This substantial improvement is attributed to our efficient kernel design.

End-to-end efficiency. *Fig.* 8 compares the end-to-end latency of *TokenSelect*, InfLLM, and standard attention across various tasks. *TokenSelect* significantly accelerates long-context inference in real-world scenarios, achieving a maximum speedup of 4.70× over standard attention and 2.28× over the SOTA long-context inference method. Moreover, *TokenSelect* demonstrates superior performance compared to both of them.

6.6 Scaling Beyond 1 Million Context Length

To further explore *TokenSelect*'s performance in extreme long-context scenarios, we design an extended benchmark with different text lengths following InfiniteBench. As illustrated in the *Fig.* 9, our *TokenSelect* demonstrates the ability to accurately capture critical information with a small token budget in contexts up to 2M tokens, underscoring its potential in more application scenarios.

7 Conclusion

In this paper, we introduces *TokenSelect*, a model-agnostic and training-free approach for efficient long-context inference and length extrapolation. *TokenSelect* addresses the two major challenges faced by LLMs in processing long texts: the context length limitation from pre-training and the computational complexity of attention. This is achieved through a novel token-level selective sparse attention mechanism. Experimental results demonstrate that *TokenSelect* can achieve up to 23.84× speedup in attention computation and up to 2.28× acceleration in end-to-end inference latency, while exhibiting superior performance across multiple long-context benchmarks. This approach significantly enhances LLMs' capability to handle long contexts, paving the way for efficient long-text processing in advancing Web applications.

References

[1] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 3119–3137. https://doi.org/10.18653/v1/2024.acl-long.172

[2] Nikhil Sharma, Q. Vera Liao, and Ziang Xiao. 2024. Generative Echo Chamber? Effect of LLM-Powered Search Systems on Diverse Information Seeking. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 1033, 17 pages. https://doi.org/10.1145/3613904.3642459

[3] Fengji Zhang, Bei Chen, Yue Zhang, Jacky Keung, Jin Liu, Daoguang Zan, Yi Mao, Jian-Guang Lou, and Weizhu Chen. 2023. RepoCoder: Repository-Level Code Completion Through Iterative Retrieval and Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 2471–2484. https://doi.org/10.18653/v1/2023.emnlp-main.151

[4] Peng Di, Jianguo Li, Hang Yu, Wei Jiang, Wenting Cai, Yang Cao, Chaoyu Chen, Dajun Chen, Hongwei Chen, Liang Chen, Gang Fan, Jie Gong, Zi Gong, Wen Hu, Tingting Guo, Zhichao Lei, Ting Li, Zheng Li, Ming Liang, Cong Liao, Bingchang Liu, Jiachen Liu, Zhiwei Liu, Shaojun Lu, Min Shen, Guangpei Wang, Huan Wang, Zhi Wang, Zhaogui Xu, Jiawei Yang, Qing Ye, Gehao Zhang, Yu Zhang, Zelin Zhao, Xunjin Zheng, Hailian Zhou, Lifu Zhu, and Xianying Zhu. 2024. CodeFuse-13B: A Pretrained Multi-lingual Code Large Language Model. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice* (Lisbon, Portugal) *(ICSE-SEIP '24)*. Association for Computing Machinery, New York, NY, USA, 418–429. https://doi.org/10.1145/3639477.3639719

[5] OpenAI. [n. d.]. Introducing OpenAI o1. https://openai.com/o1/. [Accessed 06-10-2024].

[6] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient Streaming Language Models with Attention Sinks. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=NG7sS51zVF

[7] Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2024. LM-Infinite: Zero-Shot Extreme Length Generalization for Large Language Models. arXiv:2308.16137 [cs.CL] https://arxiv.org/abs/2308.16137

[8] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, Andrea Tacchetti, Colin Gaffney, Samira Daruki, Olcan Sercinoglu, Zach Gleicher, Juliette Love, Paul Voigtlaender, Rohan Jain, Gabriela Surita, Kareem Mohamed, Rory Blevins, Junwhan Ahn, Tao Zhu, Kornraphop Kawintiranon, Orhan Firat, Yiming Gu, Yujing Zhang, Matthew Rahtz, Manaal Faruqui, Natalie Clay, Justin Gilmer, JD Co-Reyes, Ivo Penchev, Rui Zhu, Nobuyuki Morioka, Kevin Hui, Krishna Haridasan, Victor Campos, Mahdis Mahdieh, Mandy Guo, Samer Hassan, Kevin Kilgour, Arpi Vezer, Heng-Tze Cheng, Raoul de Liedekerke, Siddharth Goyal, Paul Barham, DJ Strouse, Seb Noury, Jonas Adler, Mukund Sundararajan, Sharad Vikram, Dmitry Lepikhin, Michela Paganini, Xavier Garcia, Fan Yang, Dasha Valter, Maja Trebacz, Kiran Vodrahalli, Chulayuth Asawaroengchai, Roman Ring, Norbert Kalb, Livio Baldini Soares, Siddhartha Brahma, David Steiner, Tianhe Yu, Fabian Mentzer, Antoine He, Lucas Gonzalez, Bibo Xu, Raphael Lopez Kaufman, Laurent El Shafey, Junhyuk Oh, Tom Hennigan, George van den Driessche, Seth Odoom, Mario Lucic, Becca Roelofs, Sid Lall, Amit Marathe, Betty Chan, Santiago Ontanon, Luheng He, Denis Teplyashin, Jonathan Lai, Phil Crone, Bogdan Damoc, Lewis Ho, Sebastian Riedel, Karel Lenc, Chih-Kuan Yeh, Aakanksha Chowdhery, Yang Xu, Mehran Kazemi, Ehsan Amid, Anastasia Petrushkina, Kevin Swersky, Ali Khodaei, Gowoon Chen, Chris Larkin, Mario Pinto, Geng Yan, Adria Puigdomenech Badia, Piyush Patil, Steven Hansen, Dave Orr, Sebastien M. R. Arnold, Jordan Grimstad, Andrew Dai, Sholto Douglas, Rishika Sinha, Vikas Yadav, Xi Chen, Elena Gribovskaya, Jacob Austin, Jeffrey Zhao, Kaushal Patel, Paul Komarek, Sophia Austin, Sebastian Borgeaud, Linda Friso, Abhimanyu Goyal, Ben Caine, Kris Cao, Da-Woon Chung, Matthew Lamm, Gabe Barth-Maron, Thais Kagohara, Kate Olszewska, Mia Chen, Kaushik Shivakumar, Rishabh Agarwal, Harshal Godhia, Ravi Rajwar, Javier Snaider, Xerxes Dotiwalla, Yuan Liu, Aditya Barua, Victor Ungureanu, Yuan Zhang, Bat-Orgil Batsaikhan, Mateo Wirth, James Qin, Ivo Danihelka, Tulsee Doshi, Martin Chadwick, Jilin Chen, Sanil Jain, Quoc Le, Arjun Kar, Madhu Gurumurthy, Cheng Li, Ruoxin Sang, Fangyu Liu, Lampros Lamprou, Rich Munoz, Nathan Lintz, Harsh Mehta, Heidi Howard, Malcolm Reynolds, Lora Aroyo, Quan Wang, Lorenzo Blanco, Albin Cassirer, Jordan Griffith, Dipanjan Das, Stephan Lee, Jakub Sygnowski, Zach Fisher, James Besley, Richard Powell, Zafarali Ahmed, Dominik Paulus, David Reitter, Zalan Borsos, Rishabh Joshi, Aedan Pope, Steven Hand, Vittorio Selo, Vihan Jain, Nikhil Sethi, Megha Goel, Takaki Makino, Rhys May, Zhen Yang, Johan Schalkwyk, Christina Butterfield, Anja Hauth, Alex Goldin, Will Hawkins, Evan Senter, Sergey Brin, Oliver Woodman, Marvin Ritter, Eric Noland, Minh Giang, Vijay Bolina, Lisa Lee, Tim Blyth, Ian Mackinnon, Machel Reid, Obaid Sarvana, David Silver, Alexander Chen, Lily Wang, Loren Maggiore, Oscar Chang, Nithya Attaluri, Gregory Thornton, Chung-Cheng Chiu, Oskar Bunyan, Nir Levine, Timothy Chung, Evgenii Eltyshev, Xiance Si, Timothy Lillicrap, Demetra Brady, Vaibhav Aggarwal, Boxi Wu, Yuanzhong Xu, Ross McIlroy, Kartikeya Badola, Paramjit Sandhu, Erica Moreira, Wojciech Stokowiec, Ross Hemsley, Dong Li, Alex Tudor, Pranav Shyam, Elahe Rahimtoroghi, Salem Haykal, Pablo Sprechmann, Xiang Zhou, Diana Mincu, Yujia Li, Ravi Addanki, Kalpesh Krishna, Xiao Wu, Alexandre Frechette, Matan Eyal, Allan Dafoe, Dave Lacey, Jay Whang, Thi Avrahami, Ye Zhang, Emanuel Taropa, Hanzhao Lin, Daniel Toyama, Eliza Rutherford, Motoki Sano, HyunJeong Choe, Alex Tomala, Chalence Safranek-Shrader, Nora Kassner, Mantas Pajarskas, Matt Harvey, Sean Sechrist, Meire Fortunato, Christina Lyu, Gamaleldin Elsayed, Chenkai Kuang, James Lottes, Eric Chu, Chao Jia, Chih-Wei Chen, Peter Humphreys, Kate Baumli, Connie Tao, Rajkumar Samuel, Cicero Nogueira dos Santos, Anders Andreassen, Nemanja Rakićević, Dominik Grewe, Aviral Kumar, Stephanie Winkler, Jonathan Caton, Andrew Brock, Sid Dalmia, Hannah Sheahan, Iain Barr, Yingjie Miao, Paul Natsev, Jacob Devlin, Feryal Behbahani, Flavien Prost, Yanhua Sun, Artiom Myaskovsky, Thanumalayan Sankaranarayana Pillai, Dan Hurt, Angeliki Lazaridou, Xi Xiong, Ce Zheng, Fabio Pardo, Xiaowei Li, Dan Horgan, Joe Stanton, Moran Ambar, Fei Xia, Alejandro Lince, Mingqiu Wang, Basil Mustafa, Albert Webson, Hyo Lee, Rohan Anil, Martin Wicke, Timothy Dozat, Abhishek Sinha, Enrique Piqueras, Elahe Dabir, Shyam Upadhyay, Anudhyan Boral, Lisa Anne Hendricks, Corey Fry, Josip Djolonga, Yi Su, Jake Walker, Jane Labanowski, Ronny Huang, Vedant Misra, Jeremy Chen, RJ Skerry-Ryan, Avi Singh, Shruti Rijhwani, Dian Yu, Alex Castro-Ros, Beer Changpinyo, Romina Datta, Sumit Bagri, Arnar Mar Hrafnkelsson, Marcello Maggioni, Daniel Zheng, Yury Sulsky, Shaobo Hou, Tom Le Paine, Antoine Yang, Jason Riesa, Dominika Rogozinska, Dror Marcus, Dalia El Badawy, Qiao Zhang, Luyu Wang, Helen Miller, Jeremy Greer, Lars Lowe Sjos, Azade Nova, Heiga Zen, Rahma Chaabouni, Mihaela Rosca, Jiepu Jiang, Charlie Chen, Ruibo Liu, Tara Sainath, Maxim Krikun, Alex Polozov, Jean-Baptiste Lespiau, Josh Newlan, Zeyncep Cankara, Soo Kwak, Yunhan Xu, Phil Chen, Andy Coenen, Clemens Meyer, Katerina Tsihlas, Ada Ma, Juraj Gottweis, Jinwei Xing, Chenjie Gu, Jin Miao, Christian Frank, Zeynep Cankara, Sanjay Ganapathy, Ishita Dasgupta, Steph Hughes-Fitt, Heng Chen, David Reid, Keran Rong, Hongmin Fan, Joost van Amersfoort, Vincent Zhuang, Aaron Cohen, Shixiang Shane Gu, Anhad Mohananey, Anastasija Ilic, Taylor Tobin, John Wieting, Anna Bortsova, Phoebe Thacker, Emma Wang, Emily Caveness, Justin Chiu, Eren Sezener, Alex Kaskasoli, Steven Baker, Katie Millican, Mohamed Elhawaty, Kostas Aisopos, Carl Lebsack, Nathan Byrd, Hanjun Dai, Wenhao Jia, Matthew Wiethoff, Elnaz Davoodi, Albert Weston, Lakshman Yagati, Arun Ahuja, Isabel Gao, Golan Pundak, Susan Zhang, Michael Azzam, Khe Chai Sim, Sergi Caelles, James Keeling, Abhanshu Sharma, Andy Swing, YaGuang Li, Chenxi Liu, Carrie Grimes Bostock, Yamini Bansal, Zachary Nado, Ankesh Anand, Josh Lipschultz, Abhijit Karmarkar, Lev Proleev, Abe Ittycheriah, Soheil Hassas Yeganeh, George Polovets, Aleksandra Faust, Jiao Sun, Alban Rrustemi, Pen Li, Rakesh Shivanna, Jeremiah Liu, Chris Welty, Federico Lebron, Anirudh Baddepudi, Sebastian Krause, Emilio Parisotto, Radu Soricut, Zheng Xu, Dawn Bloxwich, Melvin Johnson, Behnam Neyshabur, Justin Mao-Jones, Renshen Wang, Vinay Ramasesh, Zaheer Abbas, Arthur Guez, Constant Segal, Duc Dung Nguyen, James Svensson, Le Hou, Sarah York, Kieran Milan, Sophie Bridgers, Wiktor Gworek, Marco Tagliasacchi, James Lee-Thorp, Michael Chang, Alexey Guseynov, Ale Jakse Hartman, Michael Kwong, Ruizhe Zhao, Sheleem Kashem, Elizabeth Cole, Antoine Miech, Richard Tanburn, Mary Phuong, Filip Pavetic, Sebastien Cevey, Ramona Comanescu, Richard Ives, Sherry Yang, Cosmo Du, Bo Li, Zizhao Zhang, Mariko Iinuma, Clara Huiyi Hu, Aurko Roy, Shaan Bijwadia, Zhenkai Zhu, Danilo Martins, Rachel Saputro, Anita Gergely, Steven Zheng, Dawei Jia, Ioannis Antonoglou, Adam Sadovsky, Shane Gu, Yingying Bi, Alek Andreev, Sina Samangooei, Mina Khan, Tomas Kocisky, Angelos Filos, Chintu Kumar, Colton Bishop, Adams Yu, Sarah Hodkinson, Sid Mittal, Premal Shah, Alexandre Moufarek, Yong Cheng, Adam Bloniarz, Jaehoon Lee, Pedram Pejman, Paul Michel, Stephen Spencer, Vladimir Feinberg, Xuehan Xiong, Nikolay Savinov, Charlotte Smith, Siamak Shakeri, Dustin Tran, Mary Chesus, Bernd Bohnet, George Tucker, Tamara von Glehn, Carrie Muir, Yiran Mao, Hideto Kazawa, Ambrose Slone, Kedar Soparkar, Disha Shrivastava, James Cobon-Kerr, Michael Sharman, Jay Pavagadhi, Carlos Araya, Karolis Misiunas, Nimesh Ghelani, Michael Laskin, David Barker, Qiujia Li, Anton Briukhov, Neil Houlsby, Mia Glaese, Balaji Lakshminarayanan, Nathan Schucher, Yunhao Tang, Eli Collins, Hyeontaek Lim, Fangxiaoyu Feng, Adria Recasens, Guangda Lai, Alberto Magni, Nicola De Cao, Aditya Siddhant, Zoe Ashwood, Jordi Orbay, Mostafa Dehghani, Jenny Brennan, Yifan He, Kelvin Xu, Yang Gao, Carl Saroufim, James Molloy, Xinyi Wu, Seb Arnold, Solomon Chang, Julian Schrittwieser, Elena Buchatskaya, Soroush Radpour, Martin Polacek, Skye Giordano, Ankur Bapna, Simon Tokumine, Vincent Hellendoorn, Thibault Sottiaux, Sarah Cogan, Aliaksei Severyn, Mohammad Saleh, Shantanu Thakoor, Laurent Shefey, Siyuan Qiao, Meenu Gaba, Shuo yiin Chang, Craig Swanson, Biao Zhang, Benjamin Lee, Paul Kishan Rubenstein, Gan Song, Tom Kwiatkowski, Anna Koop, Ajay Kannan, David Kao, Parker Schuh, Axel Stjerngren, Golnaz Ghiasi, Gena Gibson, Luke Vilnis, Ye Yuan, Felipe Tiengo Ferreira, Aishwarya Kamath, Ted Klimenko, Ken Franko, Kefan Xiao, Indro Bhattacharya, Miteyan Patel, Rui Wang, Alex Morris, Robin Strudel, Vivek Sharma, Peter Choy, Sayed Hadi Hashemi, Jessica Landon, Mara Finkelstein, Priya Jhakra, Justin Frye, Megan Barnes, Matthew Mauger, Dennis Daun, Khuslen Baatarsukh, Matthew Tung, Wael Farhan, Henryk Michalewski, Fabio Viola, Felix de Chaumont Quitry, Charline Le Lan, Tom Hudson, Qingze Wang, Felix Fischer, Ivy Zheng, Elspeth White, Anca Dragan, Jean baptiste Alayrac, Eric Ni, Alexander Pritzel, Adam Iwanicki, Michael Isard, Anna Bulanova, Lukas Zilka, Ethan Dyer, Devendra Sachan, Srivatsan Srinivasan, Hannah Muckenhirn, Honglong Cai, Amol Mandhane, Mukarram Tariq, Jack W. Rae, Gary Wang, Kareem Ayoub, Nicholas FitzGerald, Yao Zhao, Woohyun Han, Chris Alberti, Dan Garrette, Kashyap Krishnakumar, Mai Gimenez, Anselm Levskaya, Daniel Sohn, Josip Matak, Inaki Iturrate, Michael B. Chang, Jackie Xiang, Yuan Cao, Nishant Ranka, Geoff Brown, Adrian Hutter, Vahab Mirrokni, Nanxin Chen, Kaisheng Yao, Zoltan Egyed, Francois Galilee, Tyler Liechty, Praveen Kallakuri, Evan Palmer, Sanjay Ghemawat, Jasmine Liu, David Tao, Chloe Thornton, Tim Green, Mimi Jasarevic, Sharon Lin, Victor Cotruta, Yi-Xuan Tan, Noah Fiedel, Hongkun Yu, Ed Chi, Alexander Neitz, Jens Heitkaemper, Anu Sinha, Denny Zhou, Yi Sun, Charbel Kaed, Brice Hulse, Swaroop Mishra, Maria Georgaki, Sneha Kudugunta, Clement Farabet, Izhak Shafran, Daniel Vlasic, Anton Tsitsulin, Rajagopal Ananthanarayanan, Alen Carin, Guolong Su, Pei Sun, Shashank V, Gabriel Carvajal, Josef Broder, Iulia Comsa, Alena Repina, William Wong, Warren Weilun Chen, Peter Hawkins, Egor Filonov, Lucia Loher, Christoph Hirnschall, Weiyi Wang, Jingchen Ye, Andrea Burns, Hardie Cate, Diana Gage Wright, Federico Piccinini, Lei Zhang, Chu-Cheng Lin, Ionel Gog, Yana Kulizhskaya, Ashwin Sreevatsa, Shuang Song, Luis C. Cobo, Anand Iyer, Chetan Tekur, Guillermo Garrido, Zhuyun Xiao, Rupert Kemp, Huaixiu Steven Zheng, Hui Li, Ananth Agarwal, Christel Ngani, Kati Goshvadi, Rebeca Santamaria-Fernandez, Wojciech Fica, Xinyun Chen, Chris Gorgolewski, Sean Sun, Roopal Garg, Xinyu Ye, S. M. Ali Eslami, Nan Hua, Jon Simon, Pratik Joshi, Yelin Kim, Ian Tenney, Sahitya Potluri, Lam Nguyen Thiet, Quan Yuan, Florian Luisier, Alexandra Chronopoulou, Salvatore Scellato, Praveen Srinivasan, Minmin Chen, Vinod Koverkathu, Valentin Dalibard, Yaming Xu, Brennan Saeta, Keith Anderson, Thibault Sellam, Nick Fernando, Fantine Huot, Junehyuk Jung, Mani Varadarajan, Michael Quinn, Amit Raul, Maigo Le, Ruslan Habalov, Jon Clark, Komal Jalan, Kalesha Bullard,

Achintya Singhal, Thang Luong, Boyu Wang, Sujeevan Rajayogam, Julian Eisenschlos, Johnson Jia, Daniel Finchelstein, Alex Yakubovich, Daniel Balle, Michael Fink, Sameer Agarwal, Jing Li, Dj Dvijotham, Shalini Pal, Kai Kang, Jaclyn Konzelmann, Jennifer Beattie, Olivier Dousse, Diane Wu, Remi Crocker, Chen Elkind, Siddhartha Reddy Jonnalagadda, Jong Lee, Dan Holtmann-Rice, Krystal Kallarackal, Rosanne Liu, Denis Vnukov, Neera Vats, Luca Invernizzi, Mohsen Jafari, Huanjie Zhou, Lilly Taylor, Jennifer Prendki, Marcus Wu, Tom Eccles, Tianqi Liu, Kavya Kopparapu, Francoise Beaufays, Christof Angermueller, Andreea Marzoca, Shourya Sarcar, Hilal Dib, Jeff Stanway, Frank Perbet, Nejc Trdin, Rachel Sterneck, Andrey Khorlin, Dinghua Li, Xihui Wu, Sonam Goenka, David Madras, Sasha Goldshtein, Willi Gierke, Tong Zhou, Yaxin Liu, Yannie Liang, Anais White, Yunjie Li, Shreya Singh, Sanaz Bahargam, Mark Epstein, Sujoy Basu, Li Lao, Adnan Ozturel, Carl Crous, Alex Zhai, Han Lu, Zora Tung, Neeraj Gaur, Alanna Walton, Lucas Dixon, Ming Zhang, Amir Globerson, Grant Uy, Andrew Bolt, Olivia Wiles, Milad Nasr, Ilia Shumailov, Marco Selvi, Francesco Piccinno, Ricardo Aguilar, Sara McCarthy, Misha Khalman, Mrinal Shukla, Vlado Galic, John Carpenter, Kevin Villela, Haibin Zhang, Harry Richardson, James Martens, Matko Bosnjak, Shreyas Rammohan Belle, Jeff Seibert, Mahmoud Alnahlawi, Brian McWilliams, Sankalp Singh, Annie Louis, Wen Ding, Dan Popovici, Lenin Simicich, Laura Knight, Pulkit Mehta, Nishesh Gupta, Chongyang Shi, Saaber Fatehi, Jovana Mitrovic, Alex Grills, Joseph Pagadora, Dessie Petrova, Danielle Eisenbud, Zhishuai Zhang, Damion Yates, Bhavishya Mittal, Nilesh Tripuraneni, Yannis Assael, Thomas Brovelli, Prateek Jain, Mihajlo Velimirovic, Canfer Akbulut, Jiaqi Mu, Wolfgang Macherey, Ravin Kumar, Jun Xu, Haroon Qureshi, Gheorghe Comanici, Jeremy Wiesner, Zhitao Gong, Anton Ruddock, Matthias Bauer, Nick Felt, Anirudh GP, Anurag Arnab, Dustin Zelle, Jonas Rothfuss, Bill Rosgen, Ashish Shenoy, Bryan Seybold, Xinjian Li, Jayaram Mudigonda, Goker Erdogan, Jiawei Xia, Jiri Simsa, Andrea Michi, Yi Yao, Christopher Yew, Steven Kan, Isaac Caswell, Carey Radebaugh, Andre Elisseeff, Pedro Valenzuela, Kay McKinney, Kim Paterson, Albert Cui, Eri Latorre-Chimoto, Solomon Kim, William Zeng, Ken Durden, Priya Ponnapalli, Tiberiu Sosea, Christopher A. Choquette-Choo, James Manyika, Brona Robenek, Harsha Vashisht, Sebastien Pereira, Hoi Lam, Marko Velic, Denese Owusu-Afriyie, Katherine Lee, Tolga Bolukbasi, Alicia Parrish, Shawn Lu, Jane Park, Balaji Venkatraman, Alice Talbert, Lambert Rosique, Yuchung Cheng, Andrei Sozanschi, Adam Paszke, Praveen Kumar, Jessica Austin, Lu Li, Khalid Salama, Wooyeol Kim, Nandita Dukkipati, Anthony Baryshnikov, Christos Kaplanis, XiangHai Sheng, Yuri Chervonyi, Caglar Unlu, Diego de Las Casas, Harry Askham, Kathryn Tunyasuvunakool, Felix Gimeno, Siim Poder, Chester Kwak, Matt Miecnikowski, Vahab Mirrokni, Alek Dimitriev, Aaron Parisi, Dangyi Liu, Tomy Tsai, Toby Shevlane, Christina Kouridi, Drew Garmon, Adrian Goedeckemeyer, Adam R. Brown, Anitha Vijayakumar, Ali Elqursh, Sadegh Jazayeri, Jin Huang, Sara Mc Carthy, Jay Hoover, Lucy Kim, Sandeep Kumar, Wei Chen, Courtney Biles, Garrett Bingham, Evan Rosen, Lisa Wang, Qijun Tan, David Engel, Francesco Pongetti, Dario de Cesare, Dongseong Hwang, Lily Yu, Jennifer Pullman, Srini Narayanan, Kyle Levin, Siddharth Gopal, Megan Li, Asaf Aharoni, Trieu Trinh, Jessica Lo, Norman Casagrande, Roopali Vij, Loic Matthey, Bramandia Ramadhana, Austin Matthews, CJ Carey, Matthew Johnson, Kremena Goranova, Rohin Shah, Shereen Ashraf, Kingshuk Dasgupta, Rasmus Larsen, Yicheng Wang, Manish Reddy Vuyyuru, Chong Jiang, Joana Ijazi, Kazuki Osawa, Celine Smith, Ramya Sree Boppana, Taylan Bilal, Yuma Koizumi, Ying Xu, Yasemin Altun, Nir Shabat, Ben Bariach, Alex Korchemniy, Kiam Choo, Olaf Ronneberger, Chimezie Iwuanyanwu, Shubin Zhao, David Soergel, Cho-Jui Hsieh, Irene Cai, Shariq Iqbal, Martin Sundermeyer, Zhe Chen, Elie Bursztein, Chaitanya Malaviya, Fadi Biadsy, Prakash Shroff, Inderjit Dhillon, Tejasi Latkar, Chris Dyer, Hannah Forbes, Massimo Nicosia, Vitaly Nikolaev, Somer Greene, Marin Georgiev, Pidong Wang, Nina Martin, Hanie Sedghi, John Zhang, Praseem Banzal, Doug Fritz, Vikram Rao, Xuezhi Wang, Jiageng Zhang, Viorica Patraucean, Dayou Du, Igor Mordatch, Ivan Jurin, Lewis Liu, Ayush Dubey, Abhi Mohan, Janek Nowakowski, Vlad-Doru Ion, Nan Wei, Reiko Tojo, Maria Abi Raad, Drew A. Hudson, Vaishakh Keshava, Shubham Agrawal, Kevin Ramirez, Zhichun Wu, Hoang Nguyen, Ji Liu, Madhavi Sewak, Bryce Petrini, DongHyun Choi, Ivan Philips, Ziyue Wang, Ioana Bica, Ankush Garg, Jarek Wilkiewicz, Priyanka Agrawal, Xiaowei Li, Danhao Guo, Emily Xue, Naseer Shaik, Andrew Leach, Sadh MNM Khan, Julia Wiesinger, Sammy Jerome, Abhishek Chakladar, Alek Wenjiao Wang, Tina Ornduff, Folake Abu, Alireza Ghaffarkhah, Marcus Wainwright, Mario Cortes, Frederick Liu, Joshua Maynez, Andreas Terzis, Pouya Samangouei, Riham Mansour, Tomasz Kępa, François-Xavier Aubet, Anton Algymr, Dan Banica, Agoston Weisz, Andras Orban, Alexandre Senges, Ewa Andrejczuk, Mark Geller, Niccolo Dal Santo, Valentin Anklin, Majd Al Merey, Martin Baeuml, Trevor Strohman, Junwen Bai, Slav Petrov, Yonghui Wu, Demis Hassabis, Koray Kavukcuoglu, Jeffrey Dean, and Oriol Vinyals. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv:2403.05530 [cs.CL] https://arxiv.org/abs/2403.05530

[9] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 Technical Report. arXiv:2407.10671 [cs.CL] https://arxiv.org/abs/2407.10671

[10] Team GLM, :, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Jingyu Sun, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. arXiv:2406.12793 [cs.CL] https://arxiv.org/abs/2406.12793

[11] Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage BERT: A Globally Normalized BERT Model for Open-domain Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 5878–5882.

[12] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for Longer Sequences. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 17283–17297. https://proceedings.neurips.cc/paper_files/paper/2020/file/c8512d142a2d849725f31a9a7a361ab9-Paper.pdf

[13] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. 2024. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems* 36 (2024).

[14] Matanel Oren, Michael Hassid, Nir Yarden, Yossi Adi, and Roy Schwartz. 2024. Transformers are Multi-State RNNs. arXiv:2401.06104 [cs.CL] https://arxiv.org/abs/2401.06104

[15] Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2024. InfLLM: Training-Free Long-Context Extrapolation for LLMs with an Efficient Context Memory. arXiv:2402.04617 [cs.CL] https://arxiv.org/abs/2402.04617

[16] Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. 2024. QUEST: Query-Aware Sparsity for Efficient Long-Context LLM Inference. In *Forty-first International Conference on Machine Learning*. https://openreview.net/forum?id=KzACYw0MTV

[17] Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. MInference 1.0: Accelerating Pre-filling for Long-Context LLMs via Dynamic Sparse Attention. arXiv:2407.02490 [cs.CL] https://arxiv.org/abs/2407.02490

[18] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. 2024. SGLang: Efficient Execution of Structured Language Model Programs. arXiv:2312.07104 [cs.AI] https://arxiv.org/abs/2312.07104

[19] Philippe Tillet, H. T. Kung, and David Cox. 2019. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages* (Phoenix, AZ, USA) *(MAPL 2019)*. Association for Computing Machinery, New York, NY, USA, 10–19. https://doi.org/10.1145/3315508.3329973

[20] Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ∞Bench: Extending Long Context Evaluation Beyond 100K Tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 15262–15277. https://aclanthology.org/2024.acl-long.814

[21] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. RULER: What's the Real Context Size of Your Long-Context Language Models? *arXiv preprint arXiv:2404.06654* (2024).

[22] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra,

Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783

[23] 01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open Foundation Models by 01.AI. arXiv:2403.04652 [cs.CL] https://arxiv.org/abs/2403.04652

[24] flashinfer ai. [n. d.]. GitHub - flashinfer-ai/flashinfer: FlashInfer: Kernel Library for LLM Serving — github.com. https://github.com/flashinfer-ai/flashinfer. [Accessed 12-10-2024].

[25] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288 [cs.CL] https://arxiv.org/abs/2307.09288

[26] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] https://arxiv.org/abs/2310.06825

[27] Yunpeng Huang, Jingwei Xu, Junyu Lai, Zixu Jiang, Taolue Chen, Zenan Li, Yuan Yao, Xiaoxing Ma, Lijuan Yang, Hao Chen, Shupeng Li, and Penghao Zhao. 2024. Advancing Transformer Architecture in Long-Context Large Language Models: A Comprehensive Survey. arXiv:2311.12351 [cs.CL] https://arxiv.org/abs/2311.12351

[28] Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, Shengen Yan, Guohao Dai, Xiao-Ping Zhang, Yuhan Dong, and Yu Wang. 2024. A Survey on Efficient Inference for Large Language Models. arXiv:2404.14294 [cs.CL] https://arxiv.org/abs/2404.14294

[29] Liang Zhao, Xiaocheng Feng, Xiachong Feng, Dongliang Xu, Qing Yang, Hongtao Liu, Bing Qin, and Ting Liu. 2024. Length Extrapolation of Transformers: A Survey from the Perspective of Positional Encoding. arXiv:2312.17044 [cs.CL] https://arxiv.org/abs/2312.17044

[30] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending Context Window of Large Language Models via Positional Interpolation. arXiv:2306.15595 [cs.CL] https://arxiv.org/abs/2306.15595

[31] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing* 568 (2024), 127063. https://doi.org/10.1016/j.neucom.2023.127063

[32] bloc97. 2023. NTK-Aware Scaled RoPE allows LLaMA models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation. Website. https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_

scaled_rope_allows_llama_models_to_have/.

[33] emozilla. 2023. Dynamically Scaled RoPE further increases performance of long context LLaMA with zero fine-tuning. Website. https://www.reddit.com/r/LocalLLaMA/comments/14mrgpr/dynamically_scaled_rope_further_increases/.

[34] Arthur Jacot, Franck Gabriel, and Clément Hongler. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems* 31 (2018).

[35] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2024. YaRN: Efficient Context Window Extension of Large Language Models. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=wHBfxhZu1u

[36] Arka Pal, Deep Karkhanis, Manley Roberts, Samuel Dooley, Arvind Sundararajan, and Siddartha Naidu. 2023. Giraffe: Adventures in Expanding Context Lengths in LLMs. arXiv:2308.10882 [cs.AI] https://arxiv.org/abs/2308.10882

[37] Shuo Yang, Ying Sheng, Joseph E. Gonzalez, Ion Stoica, and Lianmin Zheng. 2024. Post-Training Sparse Attention with Double Sparsity. arXiv:2408.07092 [cs.LG] https://arxiv.org/abs/2408.07092

[38] Junfeng Tian, Da Zheng, Yang Cheng, Rui Wang, Colin Zhang, and Debing Zhang. 2024. Untie the Knots: An Efficient Data Augmentation Strategy for Long-Context Pre-Training in Language Models. arXiv:2409.04774 [cs.CL] https://arxiv.org/abs/2409.04774

[39] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2020. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism. arXiv:1909.08053 [cs.CL] https://arxiv.org/abs/1909.08053

[40] Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. 2023. DeepSpeed Ulysses: System Optimizations for Enabling Training of Extreme Long Sequence Transformer Models. arXiv:2309.14509 [cs.LG] https://arxiv.org/abs/2309.14509

[41] Hao Liu, Matei Zaharia, and Pieter Abbeel. 2024. RingAttention with Blockwise Transformers for Near-Infinite Context. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=WsRHpHH4s0

[42] Zihang Dai*, Zhilin Yang*, Yiming Yang, William W. Cohen, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Language Modeling with Longer-Term Dependency. https://openreview.net/forum?id=HJePno0cYm

[43] Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. 2020. Compressive Transformers for Long-Range Sequence Modelling. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SylKikSYDH

[44] Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. 2022. Recurrent memory transformer. *Advances in Neural Information Processing Systems* 35 (2022), 11079–11091.

[45] Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. 2024. Leave No Context Behind: Efficient Infinite Context Transformers with Infini-attention. arXiv:2404.07143 [cs.CL] https://arxiv.org/abs/2404.07143

[46] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. arXiv:2309.06180 [cs.LG] https://arxiv.org/abs/2309.06180

[47] Huggingface. 2024. Huggingface Text Generation Inference. Website. https://github.com/huggingface/text-generation-inference.

[48] NVIDIA. 2024. TensorRT-LLM. Website. https://github.com/NVIDIA/TensorRT-LLM.

[49] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[50] Tri Dao. 2024. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. In *International Conference on Learning Representations (ICLR)*.

[51] Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. 2024. Model Tells You What to Discard: Adaptive KV Cache Compression for LLMs. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=uNrFpDPMyo

[52] Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. 2023. Scissorhands: Exploiting the Persistence of Importance Hypothesis for LLM KV Cache Compression at Test Time. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 52342–52364. https://proceedings.neurips.cc/paper_files/paper/2023/file/a452a7c6c463e4ae8fbdc614c6e983e6-Paper-Conference.pdf

[53] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. SnapKV: LLM Knows What You are Looking for Before Generation. arXiv:2404.14469 [cs.CL] https://arxiv.org/abs/2404.14469

[54] Luka Ribar, Ivan Chelombiev, Luke Hudlass-Galley, Charlie Blake, Carlo Luschi, and Douglas Orr. 2024. SparQ Attention: Bandwidth-Efficient LLM Inference. In *Forty-first International Conference on Machine Learning*. https://openreview.net/forum?id=OS5dqxmmtl

[55] Wonbeom Lee, Jungi Lee, Junghwan Seo, and Jaewoong Sim. 2024. InfiniGen: Efficient Generative Inference of Large Language Models with Dynamic KV Cache Management. arXiv:2406.19707 [cs.LG] https://arxiv.org/abs/2406.19707

[56] Hanlin Tang, Yang Lin, Jing Lin, Qingsen Han, Shikuan Hong, Yiwu Yao, and Gongyi Wang. 2024. RazorAttention: Efficient KV Cache Compression Through Retrieval Heads. arXiv:2407.15891 [cs.LG] https://arxiv.org/abs/2407.15891

[57] Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024. KIVI: A Tuning-Free Asymmetric 2bit Quantization for KV Cache. In *Forty-first International Conference on Machine Learning*. https://openreview.net/forum?id=L057s2Rq8O

[58] June Yong Yang, Byeongwook Kim, Jeongin Bae, Beomseok Kwon, Gunho Park, Eunho Yang, Se Jung Kwon, and Dongsoo Lee. 2024. No Token Left Behind: Reliable KV Cache Compression via Importance-Aware Mixed Precision Quantization. arXiv:2402.18096 [cs.LG] https://arxiv.org/abs/2402.18096

[59] Yefei He, Luoming Zhang, Weijia Wu, Jing Liu, Hong Zhou, and Bohan Zhuang. 2024. ZipCache: Accurate and Efficient KV Cache Quantization with Salient Token Identification. arXiv:2405.14256 [cs.LG] https://arxiv.org/abs/2405.14256

[60] Hao Kang, Qingru Zhang, Souvik Kundu, Geonhwa Jeong, Zaoxing Liu, Tushar Krishna, and Tuo Zhao. 2024. GEAR: An Efficient KV Cache Compression Recipe for Near-Lossless Generative Inference of LLM. arXiv:2403.05527 [cs.LG] https://arxiv.org/abs/2403.05527

[61] Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. 2024. MiniCache: KV Cache Compression in Depth Dimension for Large Language Models. arXiv:2405.14366 [cs.CL] https://arxiv.org/abs/2405.14366

[62] Zhongwei Wan, Xinjian Wu, Yu Zhang, Yi Xin, Chaofan Tao, Zhihong Zhu, Xin Wang, Siqi Luo, Jing Xiong, and Mi Zhang. 2024. D2O: Dynamic Discriminative Operations for Efficient Generative Inference of Large Language Models. arXiv:2406.13035 [cs.CL] https://arxiv.org/abs/2406.13035

[63] Yuxin Zhang, Yuxuan Du, Gen Luo, Yunshan Zhong, Zhenyu Zhang, Shiwei Liu, and Rongrong Ji. 2024. CaM: Cache Merging for Memory-efficient LLMs Inference. In *Proceedings of the 41st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 235)*, Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.). PMLR, 58840–58850. https://proceedings.mlr.press/v235/zhang24n.html

[64] Zheng Wang, Boxiao Jin, Zhongzhi Yu, and Minjia Zhang. 2024. Model Tells You Where to Merge: Adaptive KV Cache Merging for LLMs on Long-Context Tasks. arXiv:2407.08454 [cs.CL] https://arxiv.org/abs/2407.08454

[65] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

A Formal Statement and Proof of Lemma 1

Lemma 1 (Invariant Top-k Key Selection under Cosine Similarity Threshold, Formal).

Assumptions:

(1) Let $\mathbf{q}_1, \mathbf{q}_2 \in \mathbb{R}^d$ be two query vectors.
(2) Let $\{\mathbf{k}_i\}_{i=1}^N \subset \mathbb{R}^d$ be a finite set of key vectors.
(3) Let k be a positive integer such that $1 \leq k \leq N$.
(4) Define the cosine similarity between vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ as:

$$\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{ab}}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2},$$

where $\|\cdot\|_2$ denotes the Euclidean norm.
(5) Define the top-k selection function based on dot product similarity as: $\mathcal{I}(\mathbf{q}) = \arg\max_{S \subseteq \{1,2,\ldots,N\}, |S|=k} \sum_{i \in S} \mathbf{q} \cdot \mathbf{k}_i$. Assume that for any query vectors $\mathbf{q}$, the top-k set $\mathcal{I}(\mathbf{q})$ is uniquely determined.
(6) Let $\epsilon \in (0, 1]$ be a predefined threshold.

Lemma Statement: *If the cosine similarity between the two query vectors* $\mathbf{q}_1$ *and* $\mathbf{q}_2$ *satisfies*

$$\cos(\mathbf{q}_1, \mathbf{q}_2) > \epsilon,$$

then the indices of the top-k keys selected by $\mathbf{q}_1$ *and* $\mathbf{q}_2$ *are identical, i.e.,*

$$\mathcal{I}(\mathbf{q}_1) = \mathcal{I}(\mathbf{q}_2).$$

Proof: We start with the given condition:

$$\min_{1 \leq i \leq k} \mathbf{q}_1 \mathbf{k}_i - \max_{j > k} \mathbf{q}_1 \mathbf{k}_j > \eta,$$

which we aim to use to demonstrate that:

$$\min_{1 \leq i \leq k} \mathbf{q}_2 \mathbf{k}_i - \max_{j > k} \mathbf{q}_2 \mathbf{k}_j > 0.$$

To facilitate our analysis, we introduce the following notations:

$$\hat{\eta} = \frac{\eta}{\|\mathbf{q}_1\|}, \quad \hat{\mathbf{q}}_1 = \frac{\mathbf{q}_1}{\|\mathbf{q}_1\|}, \quad \hat{\mathbf{q}}_2 = \frac{\mathbf{q}_2}{\|\mathbf{q}_2\|}.$$

With these definitions, the original condition becomes:

$$\min_{1 \leq i \leq k} \hat{\mathbf{q}}_1 \mathbf{k}_i - \max_{j > k} \hat{\mathbf{q}}_1 \mathbf{k}_j > \hat{\eta},$$

and our goal transforms to showing:

$$\min_{1 \leq i \leq k} \hat{\mathbf{q}}_2 \mathbf{k}_i - \max_{j > k} \hat{\mathbf{q}}_2 \mathbf{k}_j > 0.$$

Next, let θ denote the angle between $\mathbf{q}_1$ and $\mathbf{q}_2$, $\cos\theta = \hat{\mathbf{q}}_1 \cdot \hat{\mathbf{q}}_2$. We can further define:

$$\mathbf{p}_1 = \mathbf{q}_2 - \mathbf{q}_1 \cos\theta, \quad \hat{\mathbf{p}}_1 = \frac{\mathbf{p}_1}{\|\mathbf{p}_1\|},$$

then $\sin\theta = \hat{\mathbf{p}}_1 \cdot \hat{\mathbf{q}}_2$, and

$$\hat{\mathbf{q}}_2 = \hat{\mathbf{q}}_1 \cos\theta + \hat{\mathbf{p}}_1 \sin\theta.$$

Then we have:

$$\begin{aligned}
\min_{1 \leq i \leq k} \hat{\mathbf{q}}_2 \mathbf{k}_i &= \min_{1 \leq i \leq k} (\hat{\mathbf{q}}_1 \cos\theta + \hat{\mathbf{p}}_1 \sin\theta) \mathbf{k}_i, \\
&\geq \min_{1 \leq i \leq k} \hat{\mathbf{q}}_1 \mathbf{k}_i \cos\theta + \min_{1 \leq i \leq k} \hat{\mathbf{p}}_1 \mathbf{k}_i \sin\theta, \\
&\geq \hat{\mathbf{q}}_1 \mathbf{k}_k \cos\theta - \|\mathbf{k}\|_{\max} \sin\theta,
\end{aligned}$$

and

$$\begin{aligned}
\max_{j > k} \hat{\mathbf{q}}_2 \mathbf{k}_j &= \max_{j > k} (\hat{\mathbf{q}}_1 \cos\theta + \hat{\mathbf{p}}_1 \sin\theta) \mathbf{k}_j \\
&\leq \max_{j > k} \hat{\mathbf{q}}_1 \mathbf{k}_i \cos\theta + \max_{j > k} \hat{\mathbf{p}}_1 \mathbf{k}_i \sin\theta, \\
&\leq \hat{\mathbf{q}}_1 \mathbf{k}_{p+1} \cos\theta + \|\mathbf{k}\|_{\max} \sin\theta.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\min_{1 \leq i \leq k} \hat{\mathbf{q}}_2 \mathbf{k}_i - \max_{j > k} \hat{\mathbf{q}}_2 \mathbf{k}_j &\geq \hat{\mathbf{q}}_1 \mathbf{k}_p \cos\theta - \|\mathbf{k}\|_{\max} \sin\theta \\
&\quad - (\hat{\mathbf{q}}_1 \mathbf{k}_{p+1} \cos\theta + \|\mathbf{k}\|_{\max} \sin\theta) \\
&= (\hat{\mathbf{q}}_1 \mathbf{k}_p \cos\theta - \hat{\mathbf{q}}_1 \mathbf{k}_{p+1} \cos\theta) - 2\|\mathbf{k}\|_{\max} \sin\theta \\
&\geq \hat{\eta} \cos\theta - 2\|\mathbf{k}\|_{\max} \sin\theta. \qquad (11)
\end{aligned}$$

In order to have Eqn. (11) > 0, we require

$$\begin{aligned}
&\hat{\eta} \cos\theta > 2\|\mathbf{k}\|_{\max} \sin\theta, \\
\Rightarrow\ &\frac{\sin\theta}{\cos\theta} < \frac{\hat{\eta}}{2\|\mathbf{k}\|_{\max}}, \\
\Rightarrow\ &\frac{1 - \cos^2\theta}{\cos^2\theta} < \left(\frac{\hat{\eta}}{2\|\mathbf{k}\|_{\max}}\right)^2, \\
\Rightarrow\ &\cos\theta \geq \frac{1}{\sqrt{1 + \left(\frac{\hat{\eta}}{2\|\mathbf{k}\|_{\max}}\right)^2}}.
\end{aligned}$$

This final inequality establishes a sufficient condition for the original statement to hold, thereby completing the proof.

B More Information on Dataset and Metrics

For InfiniteBench [20], we use longbook_sum_eng (En.Sum), longbook_qa_eng (En.QA), longbook_choice_eng (En.MC), longdialogue_qa_eng (En.Dia), code_debug (Code.D), math_find (Math.F), passkey (R.PK), number_string (R.Num) and kv_retrieval (R.KV) as evaluation datasets. The corresponding evaluation metrics are shown in Table 6. RULER [21] consists of various evaluation tasks: Single NIAH (needle in a haystack), Multi-keys NIAH, Multi-values NIAH, Multi-values NIAH, Multi-queries NIAH, Variable Tracking, Common Words Extraction, Frequent Words Extraction and Question Answering. The evaluation metric is match rate. For LongBench, we use all English tasks with evaluation metrics in Table 7.

C Experimental Results on LongBench

Compared to InfiniteBench and RULER, LongBench has much shorter text lengths. The 95% percentile for its lengths is 31K tokens. Considering that recent LLMs after SFT generally have context lengths of up to 32K tokens [9], LongBench is less suitable for evaluating state-of-the-art long-context inference methods. Nevertheless, as shown in Table 8, our *TokenSelect* still demonstrates superior overall performance compared to most baseline methods. It's worth noting that `Yi-1.5-6B` did not yield effective results on the SAMSum task because it failed to correctly follow instructions.

Table 6: Evaluation metrics of different datasets on InfiniteBench.

Datasets	En.Sum	En.QA	En.MC	En.Dia	Code.D	Math.F	R.PK	R.Num	R.KV
Metrics	Rouge-L-Sum	QA F1 Score	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy

Table 7: Evaluation metrics of different datasets on LongBench.

Datasets	NQA	Qasper	MFQA	HQA	2WikiMQA	Musique	GovReport	QMSum
Metrics	QA F1 Score	QA F1 Score	QA F1 Score	QA F1 Score	QA F1 Score	QA F1 Score	Rouge-L	Rouge-L
Datasets	MultiNews	TREC	TQA	SAMSum	PsgCount	PsgRetrieval	LCC	RepoBench-P
Metrics	Rouge-L	Accuracy	QA F1 Score	Rouge-L	Accuracy	Accuracy	Code Sim Score	Code Sim Score

Table 8: Comparison of different methods with different origin models on LongBench.

Methods	NQA	Qasper	MFQA	HQA	2WikiMQA	Musique	GovReport	QMSum	MultiNews
Qwen2-7B	24.24	45.42	47.79	42.76	44.38	24.16	33.80	23.78	26.17
NTK	26.25	45.94	50.76	53.20	50.31	30.83	32.75	23.21	25.94
SelfExtend	7.15	20.37	24.06	14.91	13.73	4.75	16.92	16.53	18.74
StreamLLM	19.49	42.56	39.63	42.43	44.67	15.22	31.51	20.57	26.00
InfLLM	27.47	41.44	46.99	47.47	49.29	25.62	32.68	23.10	26.77
TokenSelect	24.18	42.29	45.77	48.62	49.08	27.85	33.69	23.03	26.35
Llama-3-8B	19.85	42.36	41.03	47.38	39.20	22.96	29.94	21.45	27.51
NTK	9.90	45.35	49.41	48.86	29.22	24.56	34.31	23.82	27.27
SelfExtend	1.72	8.90	20.80	8.65	6.97	3.27	13.99	15.36	17.66
StreamLLM	20.05	42.46	39.54	43.69	37.89	19.68	29.17	21.33	27.56
InfLLM	22.64	43.70	49.03	49.04	35.61	26.06	30.76	22.70	27.57
TokenSelect	22.44	40.74	47.73	50.33	31.38	24.53	32.56	23.50	27.92
Yi-1.5-6B	17.18	32.56	39.06	36.26	39.25	16.32	30.53	20.21	26.20
NTK	0.80	35.06	29.05	7.47	24.38	0.73	13.66	6.25	25.43
SelfExtend	3.29	19.03	26.00	17.11	11.88	7.73	20.38	17.46	21.79
StreamLLM	15.05	33.27	38.31	34.91	36.92	16.33	29.38	20.02	26.14
InfLLM	17.65	36.25	45.40	41.25	35.89	16.94	30.22	20.85	26.04
TokenSelect	19.36	33.98	48.14	45.05	40.13	22.98	31.59	21.51	26.48

Methods	TREC	TQA	SAMSum	PsgCount	PsgRetrieval	LCC	RepoBench-P	**Average**
Qwen2-7B	78.50	88.77	46.33	5.50	70.00	62.40	61.95	45.37
NTK	79.50	89.51	46.03	5.50	60.00	59.36	59.69	46.17
SelfExtend	16.50	27.54	29.42	4.50	0.00	41.42	41.89	18.65
StreamLLM	75.50	87.19	46.27	3.50	27.50	61.18	61.12	40.27
InfLLM	70.50	87.51	44.53	4.00	46.50	55.08	57.53	42.90
TokenSelect	74.00	89.26	45.94	5.00	42.50	61.48	59.33	43.64
Llama-3-8B	74.00	90.50	42.30	8.50	62.50	60.83	49.14	42.46
NTK	73.00	88.74	42.51	8.87	99.50	33.62	35.04	42.12
SelfExtend	20.50	16.82	25.39	5.75	7.50	26.24	31.22	14.42
StreamLLM	73.50	90.08	41.55	5.00	49.00	60.35	48.95	40.61
InfLLM	73.50	90.91	42.43	7.17	84.00	59.88	46.48	44.46
TokenSelect	67.50	92.22	42.16	4.54	87.00	58.86	51.24	44.04
Yi-1.5-6B	71.50	48.79	0.79	3.00	28.50	57.10	52.53	32.48
NTK	40.00	12.71	1.34	0.50	3.35	54.55	37.24	18.28
SelfExtend	23.75	30.61	2.58	2.75	13.50	43.17	35.45	18.53
StreamLLM	69.00	73.36	0.82	2.50	18.50	56.37	49.05	32.49
InfLLM	71.50	71.49	1.01	4.00	10.50	56.88	46.28	33.25
TokenSelect	62.50	69.70	0.62	3.50	41.50	54.32	54.99	36.02