FASA: FREQUENCY-AWARE SPARSE ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

The deployment of Large Language Models (LLMs) faces a critical bottleneck when handling lengthy inputs: the prohibitive memory footprint of the Key Value (KV) cache. To address this bottleneck, the token pruning paradigm leverages attention sparsity to selectively retain a small, critical subset of tokens. However, existing approaches fall short, with static methods risking irreversible information loss and dynamic strategies employing heuristics that insufficiently capture the query-dependent nature of token importance. We propose FASA, a novel framework that achieves query-aware token eviction by dynamically predicting token importance. FASA stems from a novel insight into RoPE: the discovery of functional sparsity at the frequency-chunk (FC) level. Our key finding is that a small, identifiable subset of "dominant" FCs consistently exhibits high contextual agreement with the full attention head. This provides a robust and computationally free proxy for identifying salient tokens. Building on this insight, FASA first identifies a critical set of tokens using dominant FCs, and then performs focused attention computation solely on this pruned subset. Across a spectrum of long-context tasks, from sequence modeling to complex CoT reasoning, FASA consistently outperforms all token-eviction baselines and achieves near-oracle accuracy, demonstrating remarkable robustness even under constraint budgets. Notably, on LongBench-V1, FASA reaches nearly 100% of full-KV performance when only keeping 256 tokens, and achieves 2.56× speedup using just 18.9% of the cache on AIME24.

1 INTRODUCTION

Despite recent advances in Large Language Models (Dao et al., 2022; Ainslie et al., 2023; Liu et al., 2024a) in long-context processing, requirements such as repository-level code analysis (Chen et al., 2021) and document summarization (Goyal & Durrett, 2020) pose both memory and computational challenges, especially the linear growth of the KV cache. As the sequences grow, each token generation requires accessing the entire KV cache, leading to increased memory I/O latency. This memory-bound process underutilizes high-performance GPUs, ultimately limiting the overall throughput. To optimize KV cache management, previous studies have proposed mainly five directions: *token eviction* (Akhauri et al., 2025), *low-rank compression* (Chang et al., 2025; Singhania et al., 2024; Zhang et al., 2025), *quantization* (Hooper et al., 2025b; Liu et al., 2024d), *KV merging* (Wang et al., 2025b; Wan et al., 2025; Liu et al., 2024b), and *budget allocation* (Cai et al., 2025b).

Among these, an intuitive and widely explored approach is *token eviction* (LI et al., 2025; Liu et al., 2023). The rationale is that only a small subset of tokens contributes significantly to outputs, enabling the selective removal of trivial ones. Existing token eviction methods can be classified into three types: **(1)** *Static strategies* remove tokens with fixed rules (Xiao et al., 2024), therefore risking irreversible information loss; **(2)** *Adaptive strategies* either permanently evict less critical tokens (Zhang et al., 2023; Li et al., 2024) or preserve the full cache while retrieving a subset of entries (Tang et al., 2024; Ge et al., 2024). Yet such heuristic rankings provide an imperfect proxy for the truly dynamic nature of token importance; **(3)** *Learning-based strategies* (Akhauri et al., 2025; Yang et al., 2025; Chen et al., 2025) rely on a trained token predictor, suffering from poor generalization on different datasets. *Can a token predictor achieve* ***query-awareness*** *without resorting to costly training?*

In response to this question, we introduce FASA (Frequency-Aware Sparse Attention), a **training-free**, **high-granularity**, **query-aware** predictor designed to evaluate token significance during the decoding phase, in a training-free manner. The design of FASA is rooted in an intriguing observation that differential frequencies within RoPE (Su et al., 2023) induce functional sparsity among frequency chunks (FCs). Only a sparse subset of FCs, termed as dominant FCs, contribute significantly to contextual awareness, while others construct robust positional patterns. We empirically verify that these dominant FCs are sparse, universal, and task-agnostic in Section 3.3, thereby providing a robust foundation for accurately predicting token importance.

Building upon this insight, FASA employs a two-stage framework for efficient inference. The first stage, Token Importance Prediction, harnesses dominant FCs to dynamically estimate attention scores, obtaining critical tokens. At the second stage, Focused Attention Computation then performs precise and focused token generation on this reduced set. The overhead of FASA is minimal because the identification of dominant FCs is a one-time and task–invariant process. Ultimately, FASA achieves high efficiency by fetching only a small fraction of the KV cache, which significantly reduces the data transferred between memory and the processor and thereby lowers memory bandwidth consumption. The overview of FASA is in Figure 2. Grounded on the same principles above, we introduce two variants of FASA: **FASA-M** and **FASA-C**. While they differ in implementation strategies, both *achieve equivalent downstream task performance* while offering different efficiency profiles, specializing in memory and computation, respectively. Crucially, despite FASA leverages a low-rank subspace, its primary objective is the dynamic prediction of token importance, not mere dimensionality reduction. This design makes FASA orthogonal to and compatible with most other KV cache compression methods. For example, it can be seamlessly integrated with layer-wise budget allocation schemes like PyramidKV (Cai et al., 2025b).

We evaluated FASA across a range of LLMs with varying KV cache budgets, concentrating on three core tasks: long-context benchmark, long-sequence modeling, and long chain-of-thought (LongCoT) reasoning. Our method achieves performance comparable to that of full KV cache, with reduction of less than 0.7%, while consistently surpassing all baseline methods across these tasks. FASA-M provides an 8× compression of the KV cache, substantially optimizing memory usage. and FASA-C delivers 2.6× speedups, enhancing computational efficiency, with 25% of FCs selected. Our contributions are summarized as follows:

- We are the first to uncover an intriguing finding: functional sparsity at FC-level induced by RoPE.
- Leveraging the functional sparsity of FCs, we introduce FASA, a training-free framework for dynamically predicting token importance.
- We present two variants of FASA: FASA-M, optimized for settings with memory constraints, and FASA-C, designed for scenarios with computational constraints.
- Extensive experiments across three paradigm tasks demonstrate that FASA consistently achieves near-oracle accuracy in both long-context and long-generation tasks.

2 Related Works

Token Eviction. A central theme in recent KV cache optimization (Hooper et al., 2025a; Wang et al., 2025a) is the exploitation of inherent, query-dependent attention sparsity (Liu et al., 2024c; 2025; Behnam et al., 2025). Stream (Xiao et al., 2024) employs a rigid heuristic, preserving only initial and recent tokens, which invariably discards potentially crucial information from intermediate positions. SnapKV (Li et al., 2024) improves on this by introducing a one-time, prefill-stage filtering based on empirically estimated attention scores. However, the static nature of this estimation cannot adapt to the evolving relevance of tokens as generation progresses. Quest (Tang et al., 2024) offers a more dynamic solution by organizing the KV cache into pages and selectively fetching them. Despite its dynamism, its efficacy is hampered by a coarse, page-level granularity, which incurs significant overhead by forcing the retrieval of entire pages even when only a few tokens are needed.

Low-rank Compression. Another prominent paradigm for KV cache compression is low-rank approximation (Zhang et al., 2025; Dong et al., 2024), predicated on the observation that the cache's information content is concentrated in a low-dimensional subspace (Sun et al., 2025; sax, 2024; Behnam et al., 2025). For instance, SparQ (Ribar et al., 2024) employs a heuristic that selects key dimensions based on high query-vector magnitudes, a strategy that proves suboptimal due to its head-agnostic nature and its simplistic reliance on magnitude as a proxy for importance. Similarly, LoKi (Singhania et al., 2024) leverages Principal Component Analysis (PCA) to project key states into a compact subspace for efficient computation, but at the cost of significant memory overhead from storing the requisite projection matrices. In contrast, our proposed FASA circumvents these limitations by operating in-place on the KV cache, thereby incurring no auxiliary memory overhead.

3 Observation

3.1 Preliminary: Rotary Positional Encodings (RoPE)

RoPE embeds relative position information into the self-attention computation. Specifically, for a query vector $\mathbf{q}_{t_1}$ and a key vector $\mathbf{k}_{t_2}$ at positions t_1 and t_2, the attention score is formulated as $\mathbf{A}_{t_1,t_2} = (\mathbf{q}_{t_1}\mathbf{R}_{t_1})(\mathbf{k}_{t_2}\mathbf{R}_{t_2})^\top = \mathbf{q}_{t_1}\mathbf{R}_{\Delta t}\mathbf{k}_{t_2}^\top$. Due to the orthogonality, the product of $\mathbf{R}_{t_1}$ and $\mathbf{R}_{t_2}$ elegantly simplifies to a single rotation matrix parameterized solely by the relative offset $\Delta t = t_1 - t_2$.

A Frequency-Chunk Perspective on RoPE. From a frequency-domain perspective, the RoPE mechanism can be interpreted through the concept of "frequency chunks" (FCs). This framework posits that any d-dimensional vector $\mathbf{v} \in \mathbb{R}^d$ (e.g., a query and key) is partitioned into $d/2$ orthogonal 2D subspaces. We denote the i-th such subspace, or FC, as $\mathbf{v}^{[i]} = (v_{2i}, v_{2i+1})^T$. Each FC is associated with a unique base angular frequency, calculated as $\theta_i = B^{-2(i-1)/d}$ for $i \in \{1, \ldots, d/2\}$, where B is a predefined frequency base. This design establishes a direct mapping from a chunk's dimensional indices $(2i, 2i+1)$ to its rotational frequency. *Lower dimension indices (i) result in higher frequencies, which implies that the corresponding FCs rotate very quickly physically.* For a token at absolute position m, its i-th FC is rotated by an angle $m\theta_i$ through a specific 2×2 rotation matrix $\mathbf{R}_{m,\theta_i}$. The global rotation matrix $\mathbf{R}_{\Delta t}$ is block-diagonal, where each diagonal block is a 2×2 rotation matrix $\mathbf{R}_{\Delta t,\theta_i}$ and defined as $\mathbf{R}_{\Delta t} = \mathrm{Diag}(\mathbf{R}_{\Delta t,\theta_1}, \mathbf{R}_{\Delta t,\theta_2}, \ldots, \mathbf{R}_{\Delta t,\theta_{d/2}}) = \bigoplus_{i=1}^{d/2} \mathbf{R}_{\Delta t,\theta_i}$.

$$\mathbf{v}_m = \bigoplus_{k=1}^{d/2} \mathbf{v}_m^{[i]} = \bigoplus_{k=1}^{d/2} (\mathbf{v}_{2i}, \mathbf{v}_{2i+1})^T, \mathbf{R}_{m,\theta_i} = \begin{pmatrix} \cos(m\theta_i) & -\sin(m\theta_i) \\ \sin(m\theta_i) & \cos(m\theta_i) \end{pmatrix}. \tag{1}$$

3.2 Motivation and Hypothesis

Position vs. Semantics: Different Roles of FCs. The varying rotational velocities across FCs inherently lead to functional heterogeneity. This principle is substantiated by two key observations from prior literature. First, a distinct division of labor exists within RoPE (Barbero et al., 2025; Wei et al., 2025), where high-frequency FCs (in low dimensions) are primarily responsible for constructing robust positional patterns, and in contrast, low-frequency counterparts specialize in carrying the semantic information and model long-range dependencies. Second, this functional specialization is structurally reflected by a RoPE-induced concentration of high-magnitude values within specific query and key dimensions (Sun et al., 2024), reinforcing the non-uniform functional importance of FCs. This functional heterogeneity suggests that FCs can be grouped into two distinct categories:

1. **Contextual FCs:** A small, critical subset responsible for dynamic, context-specific attention. These FCs identify which tokens are semantically relevant to the current query.
2. **Structural FCs:** The remaining majority primarily injects inherent, positional attention patterns, mainly recency bias (Peysakhovich & Lerer, 2023) and attention sinks (Xiao et al., 2024).

Hypothesis: *The model's contextual awareness is overwhelmingly driven by the Contextual FCs. A few contextual FCs could replicate the contextual selection behavior of a full attention head.* If their index set is denoted as $\mathcal{I}_{\text{dom}} \subset \{1, \ldots, d/2\}$, the full attention dot product can be effectively approximated by summing only over $\mathcal{I}_{\text{dom}}$, namely $\mathbf{A}_{t_1,t_2} = \mathbf{q}_{t_1}\mathbf{R}_{\Delta t}\mathbf{k}_{t_2}^T \sum_{i\in\mathcal{I}_{\text{dom}}} \mathbf{q}_{t_1}^{[i]}\mathbf{R}_{\Delta t,\theta_i}\mathbf{k}_{t_2}^{[i]\top}$.

3.3 Quantifying Functional Sparsity

Quantifying our hypothesis of FC-level functional sparsity requires a metric to assess the "dominance" of individual FCs. Therefore, we propose the **Contextual Agreement (CA)** metric, which measures the alignment between the attention pattern from a single FC and that of the full attention head.

Formal Setup. For a query $\mathbf{q}_t \in \mathbb{R}^d$ and key matrix $\mathbf{K}_{1:t} \in \mathbb{R}^{d\times t}$ in an attention head (l, h), we define two raw score vectors: the standard **full-head scores** $\boldsymbol{\alpha}_{l,h}$ and the **single-FC scores** $\boldsymbol{\alpha}_{l,h}^{(i)}$. The latter are computed using only the 2D components of the i-th FC. These are expressed as:

$$\boldsymbol{\alpha}_{l,h}(\mathbf{q}_t, \mathbf{K}_{1:t}) = [\mathbf{q}_t\,\mathbf{R}_{t-1}\,(\mathbf{k}_0)^T, \cdots, \mathbf{q}_t\,\mathbf{R}_0\,(\mathbf{k}_t)^T]^T \tag{2}$$

$$\boldsymbol{\alpha}_{l,h}^{(i)}(\mathbf{q}_t, \mathbf{K}_{1:t}) = [\mathbf{q}_t^{[i]}\,\mathbf{R}_{t-1,\theta_i}\,{\mathbf{k}_0^{[i]}}^T, \cdots, \mathbf{q}_t^{[i]}\,\mathbf{R}_{0,\theta_i}\,{\mathbf{k}_t^{[i]}}^T]^T \tag{3}$$

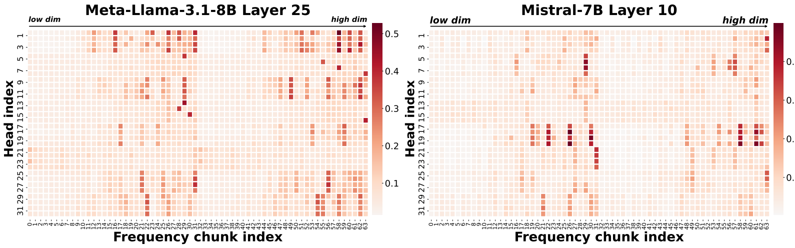


Figure 1: **Functional sparsity of FCs revealed by Contextual Agreement ($\overline{\text{CA}}$) heatmaps.** Each heatmap shows $\overline{\text{CA}}$ per FC (x-axis) across all heads (y-axis). A few "dominant" FCs (bright vertical bands) consistently capture contextual information across attention heads. Results on Qasper ($\mathcal{K} = 256$); see Appendix B.

Metric Definition. The **CA** score, $\text{CA}_{\mathcal{K}}^{l,h,i}$, quantifies the agreement between the full-head $\boldsymbol{\alpha}_{l,h}$ and single-FC $\boldsymbol{\alpha}_{l,h}^{(i)}$ scores by measuring the normalized intersection of their top-$\mathcal{K}$ token index sets:

$$\text{CA}_{\mathcal{K}}^{l,h,i}(q_t, \mathbf{K}_{1:t}) = [\text{TopK-I}(\boldsymbol{\alpha}_{l,h}(q_t, \mathbf{K}_{1:t}), \mathcal{K}) \cap \text{TopK-I}(\boldsymbol{\alpha}_{l,h}^{(i)}(q_t, \mathbf{K}_{1:t}), \mathcal{K})]/\mathcal{K}, \tag{4}$$

where the operator $\text{TopK-I}(\boldsymbol{\alpha}, \mathcal{K})$ retrieves the top-$\mathcal{K}$ values of a vector $\boldsymbol{\alpha}$. To assess an FC's importance robustly, we compute its mean CA score, by averaging across several samples from a specific dataset. Figure 1 reveals the distinct functional contribution of each FC across all heads.

Sparse and Universal $\mathcal{I}_{\textbf{dom}}$. Empirical analysis reveals three properties: **(1) *Sparsity*:** a small subset of FCs (dominant FCs) exhibits disproportionately high agreement with full attention patterns. Conversely, the CA scores for the vast majority of other FCs are negligible (typically < 0.1); **(2) *Universality*:** The functional sparsity is widely observed across Llama, Mistral, and Qwen, and model scales from 3B to 32B (Appendix B.1); **(3) *Task-Invariance:*** The set of dominant FCs is largely task-agnostic. As shown in Figure 15, the saliency maps derived from tasks such as QA and summarization are consistent, suggesting that the functional roles of FCs are intrinsic to the RoPE's mechanics, rather than being task-specific adaptations.

Table 1: Compound CA scores under varying number of selected FCs (F) and KV cache budgets (K). Each head has 64 FCs in total.

K / $\lvert\mathcal{I}_{dom}\rvert$	64	256	512	768	1024	2048
Random	2.0	3.6	6.4	19.1	25.5	51.1
Stream	34.4	26.8	24.4	26.5	30.7	53.9
SnapKV	37.9	40.9	41.9	45.4	49.5	66.6
$F = \mathbf{8}$ (1/8)	43.0	49.4	54.3	58.8	62.6	76.1
$F = \mathbf{10}$	46.4	52.1	56.6	61.1	64.8	77.5
$F = \mathbf{12}$	49.7	54.7	58.9	63.4	66.8	79.0
$F = \mathbf{14}$	52.4	56.9	60.9	65.2	68.5	80.2
$F = \mathbf{16}$ (1/4)	55.3	59.7	62.8	66.9	70.1	81.4

Quantitative Evidence about the property of Sparsity & Universality & Task-Invariance **For sparsity**, as shown in Table 16, We quantitatively analyzed the proportion of dominant FCs (defined as CA > 0.4). We found they account for less than 1% of all FCs, while non-dominant FCs with low CA scores comprise approximately 90% or more. This sparsity pattern holds universally. We confirmed its existence across different architectures (Llama, Qwen, Mistral, R1 models) and scales (3B to 32B), which strongly supports the **universality** claim. For **Task-Invariance**, our analysis reveals a remarkably high degree of overlap on dominant FCs, which consistently exceeds 70% across all tested models and tasks in Table 17, when using varying calibration datasets.

Reconstructing Functionality from $\mathcal{I}_{\textbf{dom}}$. The analysis above supports that the functionality of a full attention head can be reconstructed using only its most dominant F components $\mathcal{I}_{\text{dom}}^{l,h} = \text{TopK-I}(\{\text{CA}_{\mathcal{K}}^{l,h,i} \mid 0 \le f < d/2\}, F)$. Therefore, we measure the collective efficacy of this subset using a compound CA score, $\text{CA}_{K}^{l,h,\mathcal{I}_{\text{dom}}}$, and present the results in Table 1. For comparison, we benchmark against token-eviction methods, which serve to emphasize the capability of predicting token importance. Our method demonstrates remarkable efficiency: with just 1/8 of the components selected under a tight budget 64, $\mathcal{I}_{\text{dom}}$ achieves an accuracy of 43%, surpassing the strong baseline SnapKV (Li et al., 2024) by an average of 10.3% across all budget levels.

4 METHOD

Grounded in the functional sparsity of FCs, our training-free framework FASA employs a two-stage, coarse-to-fine strategy to circumvent the prohibitive cost of full self-attention. First, the

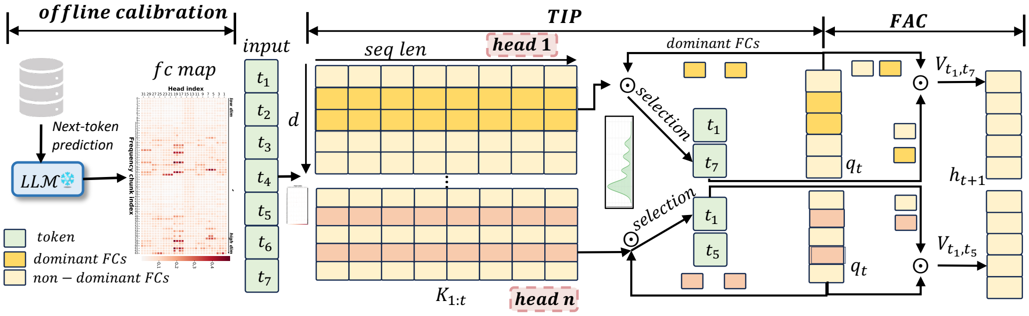


Figure 2: Method Overview of FASA. First, the **TIP** stage leverages only dominant FCs to efficiently estimate token importance and select a critical subset of tokens. Then, the **FAC** stage performs full-dimensional attention exclusively on this reduced subset to generate the next token. See discussion about design in Appendix E.2.

Token Importance Predictor (TIP) stage utilizes a computationally frugal proxy, defined by a pre-calibrated set of dominant FCs, $\mathcal{I}_{\text{dom}}$, to efficiently identify a small subset of contextually salient tokens. Subsequently, the **Focused Attention Computation (FAC)** stage performs a full-fidelity attention computation exclusively on this salient subset, preserving high generation fidelity while drastically mitigating the computational and memory overhead of standard attention.

4.1 Token Importance Predictor (TIP)

The TIP stage operates on the principle that dominant frequencies are an efficient proxy for token importance, where the dominant indices $\mathcal{I}_{dom}$ are identified via a one-time offline calibration.

Offline Calibration: Identifying $\mathcal{I}_{dom}$. The objective of the offline calibration is to identify a small, head-specific set of *dominant frequencies*, $\mathcal{I}_{\text{dom}}^{l,h}$, for each attention head (l, h). We formulate this process as a search problem over frequency indices. Given a small calibration dataset Ω and a target size N_{tip}, our goal is to find the subset of FCs of cardinality N_{tip} that maximizes the expected average of CA scores. The objective is defined as:

$$\mathcal{I}_{\text{dom}}^{l,h} = \underset{\mathcal{I}\subseteq\{0,\ldots,d/2-1\},|\mathcal{I}|=N_{tip}}{\operatorname{argmax}} \mathbb{E}_{\mathbf{q},\mathbf{K}\sim\Omega}\left[\sum_{i\in\mathcal{I}} \text{CA}_{\mathcal{K}}^{l,h,i}(\mathbf{q},\mathbf{K})\right]. \tag{5}$$

This calibration is a highly efficient, one-time offline process because the resulting $\mathcal{I}_{\text{dom}}$ is empirically found to be task-agnostic and can be robustly identified from a minimal number of samples. Its associated computational cost is negligible. The detailed algorithm is provided in Algorithm 1.

Online Prediction: Importance Scoring via Frequency Subspace Aggregation. During the online prediction phase at a given decoding step t, we leverage the pre-calibrated set of dominant frequencies, $\mathcal{I}_{\text{dom}}^{l,h}$, to efficiently estimate token importance in a training-free manner. Conceptually, the full attention score for a query $\mathbf{q}_t$ and keys $\mathbf{K}_{1:t}$ can be decomposed into a sum of contributions from all $d/2$ frequency components: $\boldsymbol{\alpha}^{l,h}(\mathbf{q}_t, \mathbf{K}_{1:t}) = \sum_{i=0}^{d/2-1} \boldsymbol{\alpha}^{l,h,i}(\mathbf{q}_t, \mathbf{K}_{1:t})$. Instead of performing this computationally expensive summation, our method constructs an *importance score vector* $\mathbf{S}_t^{l,h}$, by exclusively aggregating the contributions from the pre-identified dominant frequencies, i.e., $\mathbf{S}_t^{l,h} \triangleq \sum_{i\in\mathcal{I}_{\text{dom}}^{l,h}} \boldsymbol{\alpha}^{l,h,i}(\mathbf{q}_t, \mathbf{K}_{1:t})$. This formulation strategically bypasses computation for non-dominant frequencies. Finally, based on these scores, we identify the set of top-N_{fac} most important token indices, $\mathcal{T}_t$, for the subsequent FAC stage: $\mathcal{T}_t = \text{TopK-I}(\mathbf{S}_t^{l,h}, N_{fac})$.

4.2 Focused Attention Computation (FAC)

Following the identification of the contextually important token set $\mathcal{T}_t$ by the TIP module, this stage executes an attention computation on $\mathcal{T}_t$, enabling the model to concentrate its computational resources on the most salient parts of the context. Specifically, for the current query vector $\mathbf{q}_t$ at decoding step t, instead of using the full key and value matrices $(\mathbf{K}_{1:t}, \mathbf{V}_{1:t})$ from the entire past context, we first gather the keys and values corresponding to the indices in $\mathcal{T}_t$:

$$\mathbf{K}_{\mathcal{T}_t} = \text{Gather}(\mathbf{K}_{1:t}, \mathcal{T}_t), \quad \mathbf{V}_{\mathcal{T}_t} = \text{Gather}(\mathbf{V}_{1:t}, \mathcal{T}_t) \tag{6}$$

where the Gather$(\cdot)$ operation selects the rows from the original matrices specified by the index set $\mathcal{T}_t$. The attention scores for each head (l, h) are then computed using only these selected keys. The final output vector for the head is subsequently produced by weighting the selected value vectors:

$$\hat{\boldsymbol{\alpha}}_{\text{FAC}}^{l,h} = \text{Softmax}\left(\mathbf{q}_t \mathbf{K}_{\mathcal{T}_t}{}^T / \sqrt{d}\right), \quad \mathbf{O}_t^{l,h} = \hat{\boldsymbol{\alpha}}_{\text{FAC}}^{l,h} \mathbf{V}_{\mathcal{T}_t} \tag{7}$$

Critically, the original absolute positions of the tokens in $\mathcal{T}_t$ are preserved. This directly maintains the integrity of their position embeddings and the vital spatial information they encode, preventing the performance degradation associated with positional distortion. In essence, the FAC stage functions as a high-fidelity computational filter, restricting full-precision attention to the most salient tokens to achieve a compelling balance between computational efficiency and predictive accuracy.

4.3 Two Implementations of FASA

We introduce two specialized, hardware-aware variants of FASA that offer a trade-off between memory and speed: (1). **FASA-M (Memory-Optimized)** minimizes its GPU memory footprint by strategically offloading the value cache and non-dominant key components to CPU memory, making it ideal for VRAM-constrained environments. To mitigate the latency from CPU-GPU data transfer, this approach can be effectively paired with prefetching techniques. (2) **FASA-C (Computation-Optimized)** prioritizes inference speed by retaining the full cache on-GPU but accessing only a sparse subset of key states, drastically reducing memory I/O for significant acceleration. (See Appendix E.1 for details and memory analysis of FASA-M).

4.4 Efficiency Analysis of FASA

Computational Analysis. At the generation step t, the complexity of computing $\mathbf{q}_t\mathbf{K}_{1:t}^{\mathbf{T}}$ is $\mathcal{O}(td)$ and the complexity of multiplying the value states with attention scores is $\mathcal{O}(td)$ per head. For FASA, (1) the complexity of the **TIP** stage is $\mathcal{O}(2tN_{\text{tip}})$ (each FC takes up 2 dimensions), since this stage operates in low-dimensional subspaces, and (2) the **FAC** stage performs attention on a reduced set of N_{fac} tokens, leading to a complexity of $\mathcal{O}(N_{fac}d)$. Additionally, the detection of dominant frequencies $\mathcal{I}_{dom}$ is offline, one-time, and applicable for various tasks and the burdens from this part could be neglected. Assuming the complexity of selecting the top-k tokens is small, the overall complexity of FASA is $\mathcal{O}(2tN_{tip} + 2N_{fac}d)$. The theoretical speedup at decoding stage is in Equation 8.

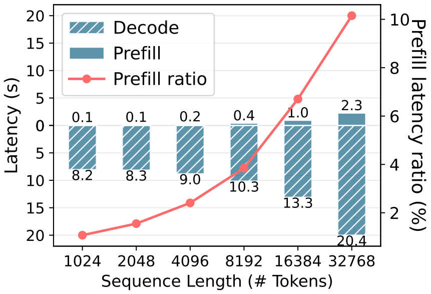


Figure 3: Decoding latency dominates total latency in auto-regressive generation.

$$\text{Speedup} = \frac{2td}{2tN_{tip} + 2N_{fac}d} = \frac{1}{N_{tip}/d + N_{fac}/t}, \text{Speedup} \approx \frac{d}{N_{tip}} \; if \; N_{fac} \ll t \tag{8}$$

Memory Movement Reduction. The auto-regressive decoding stage is notoriously memory-bound, as requiring loading the entire KV cache, creating a significant latency bottleneck. This is confirmed in Figure 3, where decoding constitutes 90% of the total latency at a 32K context. FASA, directly mitigates this bottleneck by drastically reducing memory traffic. At a decoding step t, standard attention loads $2tm$ bytes from the KV cache (with m as the byte size per state vector) while FASA accesses only $t(2N_{tip}/d * m)$ bytes (only keys) for the TIP and $2N_{\text{fac}}m$ bytes for the FAC. The fraction that FASA must load is therefore: $(2tmN_{tip}/d + 2N_{fac}m)/2tm = N_{tip}/d + N_{fac}/t \approx N_{tip}/d(N_{fac} \ll t)$, which alleviates the memory-bound constraint of long-context decoding.

5 Experiments

5.1 Experimental Setting

Baselines and Models. To comprehensively evaluate FASA's performance, we benchmark it against into two groups of robust baselines: **(1) State-of-the-art methods:** We compare against leading token eviction methods in efficient KV cache management, including Stream (Xiao et al., 2024),

Table 2: Performance of FASA on diverse models on LongBench-V1 benchmarks. For baselines, we retain constant token budget (256) and 25% FCs for FASA. †FKV and Oracle are full and look-ahead upper bounds.

	Method	Single-Doc QA			Multi-Doc QA			Summarize			Summarize		Synthetic		Code		
		NQA	Qasp	MF-en	Hqa	2Wiki	Musi	GovR	Qsum	Mult	Trec	Tqa	Pcnt	Pre	Lcc	RB-P	AVG.
Llama3.2-3B	FKV†	26.0	40.7	50.4	32.2	29.6	15.1	33.5	22.9	25.3	71.5	88.9	3.5	87.8	52.0	54.2	42.2
	Oracle†	26.6	41.2	49.8	31.9	29.9	16.2	32.6	22.2	25.0	71.5	89.3	3.5	88.0	53.7	54.4	42.4$\uparrow_{0.2}$
	Quest	8.7	19.5	23.6	12.9	15.9	6.5	23.3	18.1	25.1	34.5	52.9	6.5	38.3	53.7	43.6	25.5$\downarrow_{16.7}$
	Stream	13.2	19.7	23.6	18.1	22.7	7.8	18.2	17.9	17.9	49.0	83.7	3.5	85.7	49.3	45.9	31.8$\downarrow_{10.4}$
	SnapKV	23.5	28.9	45.6	17.7	22.9	11.8	21.7	20.9	21.1	61.0	88.5	3.5	88.0	50.7	48.6	37.0$\downarrow_{5.2}$
	FASA	**25.6**	**38.9**	**49.9**	**29.7**	**31.2**	**14.8**	**28.0**	**24.2**	**26.1**	**71.5**	**89.2**	3.6	86.9	53.2	**50.5**	**41.5**$\downarrow_{0.7}$
Qwen2.5-7B	FKV	24.2	43.5	52.1	55.9	46.9	28.6	31.8	23.1	23.9	71.5	89.3	7.5	92.0	60.2	66.5	47.8
	Oracle	24.4	43.0	52.3	57.8	46.9	30.1	31.6	23.9	24.1	72.5	89.7	8.0	100.0	60.5	65.3	48.7$\uparrow_{0.9}$
	Quest	9.1	24.5	30.4	24.7	24.1	8.8	26.8	19.9	24.4	41.8	66.7	4.4	77.6	46.5	42.0	31.4$\downarrow_{16.4}$
	Stream	18.1	24.2	26.5	41.2	36.4	17.3	18.4	18.3	15.4	45.0	82.9	8.5	24.0	49.6	52.2	31.9$\downarrow_{15.9}$
	SnapKV	26.6	36.0	50.8	55.6	43.8	26.5	21.9	21.9	19.3	58.0	86.2	8.0	98.5	55.6	60.6	42.6$\downarrow_{5.2}$
	FASA	**28.3**	**43.8**	**51.9**	**57.4**	**46.0**	**30.1**	**31.2**	**22.8**	24.3	**72.0**	**89.4**	**8.0**	**99.5**	**60.3**	**64.0**	**47.9**$\uparrow_{0.1}$
Mistral-7B-v0.3	FKV†	29.1	41.6	52.9	49.4	39.5	29.1	34.8	25.7	27.8	76.0	88.6	5.5	98.0	58.4	59.7	47.4
	Oracle†	31.0	40.2	52.4	50.3	39.4	28.8	34.0	25.74	27.2	76.0	89.4	5.0	98.0	59.3	61.0	47.9$\uparrow_{0.5}$
	Quest	15.7	30.7	41.0	37.4	27.1	11.9	29.3	21.3	26.6	57.0	80.7	5.0	85.5	56.9	53.0	38.6$\downarrow_{8.8}$
	Stream	11.8	15.3	20.9	32.1	27.1	10.6	20.2	17.3	20.1	44.5	69.0	1.6	3.2	56.5	49.8	26.7 $\downarrow_{20.7}$
	SnapKV	25.5	32.6	53.7	48.4	37.3	25.9	22.7	23.6	23.1	62.5	89.4	6.5	94.5	57.3	57.0	44.0$\downarrow_{3.4}$
	FASA	**29.9**	**42.3**	**53.7**	**51.1**	**39.1**	**28.7**	**34.0**	**24.8**	**28.2**	**76.0**	**89.4**	5.0	**98.0**	**57.8**	**58.0**	**47.8**$\uparrow_{0.4}$
Llama3.1-8B	FKV†	30.0	45.3	55.6	55.8	43.7	30.2	35.1	25.4	27.0	72.5	91.7	7.1	99.5	63.0	56.3	48.7
	Oracle†	30.3	44.5	55.0	54.9	44.6	32.0	34.8	25.1	26.9	72.5	91.5	7.0	99.5	63.3	57.4	48.7$\downarrow_{0.0}$
	Quest	13.7	33.1	38.4	35.8	32.2	12.8	26.5	20.9	26.7	38.0	65.6	3.8	95.0	52.5	45.7	35.4$\downarrow_{13.3}$
	Stream	21.9	23.4	31.8	45.1	36.7	24.3	20.0	21.0	19.3	45.5	87.9	6.9	99.5	59.4	49.1	38.8$\downarrow_{9.9}$
	SnapKV	27.5	34.5	51.6	52.3	44.3	28.3	23.9	24.0	22.7	62.5	90.9	7.5	99.5	60.1	52.6	45.0$\downarrow_{3.7}$
	FASA	**29.3**	**43.7**	**54.1**	**54.8**	43.9	**30.8**	**33.5**	**24.7**	**27.0**	**72.0**	**91.1**	**7.5**	**99.5**	**61.8**	**52.7**	**48.2**$\downarrow_{0.5}$
Qwen2.5-14B-1M	FKV†	28.7	46.2	53.8	65.2	64.5	43.6	43.5	23.3	22.7	80.5	89.5	11.0	100.0	32.3	37.5	50.3
	Oracle†	28.5	46.3	54.3	64.3	63.6	44.7	31.5	22.9	22.7	81.0	88.4	10.0	100.0	33.6	39.7	49.4$\downarrow_{0.9}$
	Quest	14.5	31.9	39.1	38.8	36.6	16.2	16.2	20.1	25.2	43.5	72.7	10.0	88.8	35.0	34.0	34.9$\downarrow_{15.4}$
	Stream	19.6	26.9	29.4	46.5	48.3	29.6	17.8	18.4	15.0	46.5	82.5	12.5	72.1	28.7	31.2	35.3$\downarrow_{15.0}$
	SnapKV	26.3	40.5	51.2	63.2	62.2	43.3	22.5	22.0	18.3	63.5	87.5	11.5	100.0	30.4	36.0	45.9$\downarrow_{4.4}$
	FASA	**27.2**	**45.5**	**54.5**	**64.4**	**63.9**	**44.5**	**30.4**	**22.8**	21.9	**80.0**	**87.5**	**15.5**	**100.0**	30.5	**36.1**	**49.2**$\downarrow_{1.1}$

SnapKV (Li et al., 2024), RKV (Cai et al., 2025a), Quest (Tang et al., 2024), H2O (Zhang et al., 2023); **(2) Upper bounds:** two theoretical bounds, FKV, which represents standard inference with the complete, uncompressed KV cache, serving as the absolute performance ceiling due to no information loss, and Oracle, a more pragmatic upper bound for eviction-based methods, assuming ideal knowledge to retain only the most critical tokens based on full-head scores. Our experiments span a variety of cutting-edge architectures and model sizes, specifically Llama (Touvron et al., 2023), Mistral (Jiang et al., 2023), and Qwen (Bai et al., 2023).

Evaluation Benchmarks. To rigorously assess the capabilities of FASA across diverse long-context scenarios, we conduct comprehensive evaluations spanning three paradigms: (1) **Long-context understanding:** We use diverse, real-world tasks from LongBench V1 (Bai et al., 2024) to assess the ability to identify critical information within lengthy contexts. (2) **Long-Sequence Modeling:** We measure perplexity on PG-19 (Rae et al., 2019), WikiText (Merity et al., 2017), and C4 (Raffel et al., 2019) datasets to evaluate generative fidelity over long dependencies. (3) **Long-CoT Reasoning:** To test performance in long-generation scenarios, we evaluate on complex mathematical reasoning tasks from MATH500 (Hendrycks et al., 2021) and AIME24 (MAA, 2024) on R1-distilled LLMs.

5.2 Performance Comparison on Long-context Tasks.

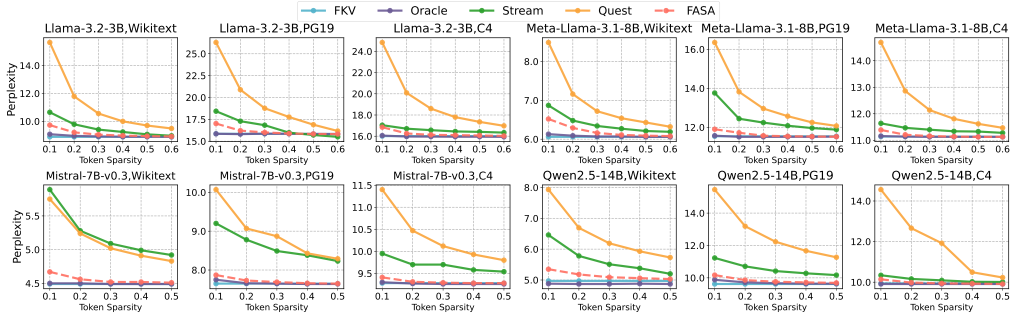


Figure 4: Perplexity results of FASA in comparison with FKV, Oracle, Stream, and Quest on Wikitext (**top**), PG19 (**middle**), and C4 corpus (**bottom**). Token sparsity indicates the retained ratio of tokens.

Table 3: Performance and output length of FASA compared to baseline models on the MATH500 and AIME24 $N_{tip} = 16$. AIME24 results are reported as pass@1, based on 16 responses per question. PREF* and DEC* denote the prefill and decoding lengths, respectively. †FKV and Oracle are full and look-ahead upper bounds.

Methods	MATH500							AIME24							
	Fixed Budget				Len Stats			Fixed Budget					Len Stats		
	300	500	700	1000	PREF*	DEC*	TOTAL.	500	1000	1500	2000	2500	PREF*	DEC*	TOTAL.
DeepSeek-R1-Distill-Llama-8B															
FKV†	72.4	-	-	72.4	127	2977	3104	43.9	-	-	-	43.9	161	13231	13392
Oracle†	70.4	72.6	74.2	71.8		3195	3321	30.0	36.7	37.3	39.3	36.0		15638	15799
H2O	6.8	33.0	53.87	42.8		8244	8370	0.7	4.7	11.3	14.0	20.0		21099	21260
Stream	9.6	24.6	40.4	47.4		3520	3647	0.0	3.3	8.0	10.7	15.3		10191	10352
SnapKV	21.6	32.6	46.8	54.6		7047	7174	4.0	8.0	16.0	23.3	29.1		17359	17520
RKV	24.0	39.4	49.2	57.0		7005	7132	6.7	10.7	14.0	21.7	23.3		22916	23077
FASA	62.2	68.8	69.4	71.8		3171	3298	20.6	34.4	40.2	35.8	38.0		17166	17327
DeepSeek-R1-Distill-Qwen-14B															
FKV†	92.4	-	-	92.4	127	2784	2914	66.6	-	-	-	66.6	165	11039	11204
Oracle†	92.2	92.4	92.4	92.2		2985	3112	67.9	66.7	67.3	70.7	67.3		11546	11711
H2O	29.6	50.2	62.8	77.0		3413	3540	5.3	20.5	37.3	46.0	52.7		9519	9684
Stream	27.8	44.0	57.8	64.4		2801	2928	2.0	4.0	16.7	22.7	29.3		8468	8633
SnapKV	34.2	55.8	69.4	79.4		3586	3713	10.0	23.3	40.0	46.0	52.7		11922	12083
RKV	57.8	74.0	80.8	86.4		3865	3992	20.7	30.0	46.7	55.4	62.0		16274	16439
FASA	86.6	88.8	90.2	91.2		3139	3266	54.0	60.6	59.3	62.7	63.3		11553	11709
DeepSeek-R1-Distill-Qwen-32B															
FKV†	92.6	-	-	92.6	127	2717	2846	72.8	-	-	-	72.8	156	10461	10626
Oracle†	92.4	91.4	91.4	91.2		2886	3013	68.0	70.1	70.0	76.7	69.2		11545	11710
H2O	47.2	50.0	68.3	74.4		3841	3968	6.7	16.7	38.4	45.6	55.6		10904	11069
Stream	43.6	57.6	65.6	73.4		2773	2900	0.7	6.7	18.7	23.3	24.7		10732	10897
SnapKV	49.6	66.0	74.8	80.8		3704	3831	10.0	23.3	40.0	46.0	52.7		13650	13815
RKV	75.0	72.2	78.4	83.6		4229	4356	14.7	32.7	43.3	55.3	61.3		18078	18243
FASA	86.4	90.2	90.2	91.2		2887	3014	60.7	62.0	66.3	70.0	73.2		11735	11891

FASA achieves near-lossless performance under various budgets. FASA consistently outperforms all baselines across various budgets (Appendix D.1 and 5), preserving contextual integrity even under extreme compression (Table 2). In stark contrast, existing token-eviction methods suffer catastrophic performance degradation; for instance, Quest's accuracy plummets by 13.4% on NarrativeQA, underscoring their inability to retain critical information. Remarkably, under extreme budgets, FASA occasionally surpasses the FKV baseline (e.g., on Mistral-7B). We attribute this phenomenon to the mitigation of attentional distraction from irrelevant tokens. This hypothesis is corroborated by the Oracle baseline, which also outperforms FKV sometimes, thereby validating our frequency-chunk-based framework's efficacy in precisely identifying semantically pivotal regions.

FASA models complex long-term dependencies. We simulate a *token-by-token* decoding process wherein the eviction strategy is iteratively applied before token prediction. The fixed-rule approach of Stream (Xiao et al., 2024), which relies on "attention sinks," severely compromises its ability to capture long-range dependencies, leading to a drastic increase in perplexity as shown in Figure 4. Similarly, Quest's coarse, page-level granularity prevents it from adaptively retaining critical, non-contiguous tokens. In contrast, FASA's fine-grained, query-dependent mechanism accurately identifies salient tokens, achieving performance comparable to FKV, even under aggressive compression.

FASA excels at long-CoT reasoning. The chain of thought in long-form reasoning is a fragile thread, requiring the preservation of dynamically shifting "thought traces", a thread that prominent baselines consistently sever. As shown in Table 3, their static compression heuristics, blind to the evolving importance of tokens, lead to a precipitous drop in performance. On R1-Llama, SnapKV's accuracy collapses to 21.6, a stark contrast to the FKV's 72.4, demonstrating a fundamental failure to sustain the very logical dependencies required for reasoning. Conversely, FASA operates with surgical precision. It surpasses not only standard baselines but also R-KV, a highly specialized method for CoT compression. It achieves an impressive 86.4% accuracy on a scant 10% context budget, narrowly trailing the 92.6% FKV upper bound. This feat cements its status as a superior framework, one that can navigate the intricate web of complex reasoning without severing the essential threads of logic.

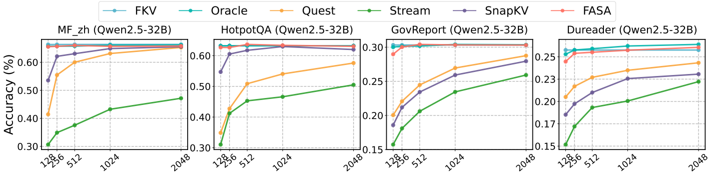


Figure 5: FASA under various token budgets ($N_{tip} = 16$).

5.3 IN-DEPTH ANALYSIS

Effect on Generation Length. A neglected aspect of compression methods is the impact on output length. Some compression methods, like H2O, induce generative verbosity, imposing an overlooked computational burden (Table 3). Conversely, others, such as Stream, prematurely terminate generation,

Table 4: Compatibility of FASA.

Budget	256	512	1024	2048
		Qasp.		
FASA	43.7	44.0	44.7	45.7
+PyKV	44.4$_{\uparrow 0.7}$	44.5$_{\uparrow 0.5}$	45.8$_{\uparrow 1.1}$	45.8$_{\uparrow 0.1}$
		Lcc		
FASA	61.8	63.4	64.4	64.8
+PyKV	62.2$_{\uparrow 0.4}$	63.6$_{\uparrow 0.2}$	64.7$_{\uparrow 0.3}$	64.9$_{\uparrow 0.1}$

Table 5: Ablation on $\mathcal{K}$.

$\mathcal{K}$	Token Budget					AVG.
	128	**256**	**512**	**1024**	**2048**	
128	42.5	43.6	**44.9**	**45.7**	**45.6**	**44.5**
256	42.6	43.7	44.0	44.7	45.3	44.1
512	41.9	43.5	43.7	44.9	45.3	43.9
1024	42.2	44.2	44.3	44.7	45.0	44.1

Table 6: Ablation of offline calibration.

Offline	S-Doc QA			M-Doc QA		
	2Wiki	**Musi**	**Hqa**	**Qasp.**	**MF_en**	**Nqa**
Base	43.7	30.2	55.8	45.3	55.6	29.9
Nqa	44.5	31.6	55.0	44.2	55.8	29.2
Qasp.	43.0	31.0	54.1	44.0	54.6	29.1
Musi	43.8	30.8	55.1	44.8	54.6	29.6
Self	43.5	30.8	55.3	43.9	54.4	29.2
CV	.014	.012	.010	.009	.011	.007

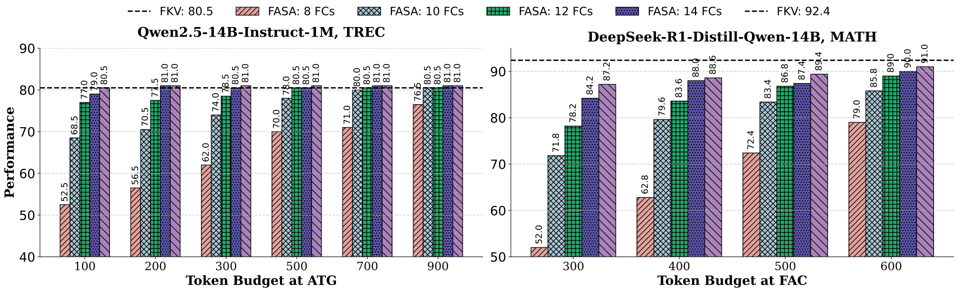


Figure 6: Evaluation of FASA on TREC (left) and MATH (right) datasets. The plots show the synergistic effects under varying numbers of selected FCs and different token budgets.

which truncates valid reasoning and degrade performance. In contrast, FASA maintains output lengths nearly identical to the FKV while preserving high performance, demonstrating a superior balance.

Compatiblility of FASA. By design, FASA is orthogonal to and synergistic with other KV cache optimization paradigms. We demonstrate this by integrating it with PyramidKV (Cai et al., 2025b), which allocates varied budgets across layers. While PyramidKV determines how many tokens to keep per layer, FASA decides which tokens are most critical. As shown in Table 4, this complementary pairing yields consistent performance gains, confirming FASA's high compatibility and modularity.

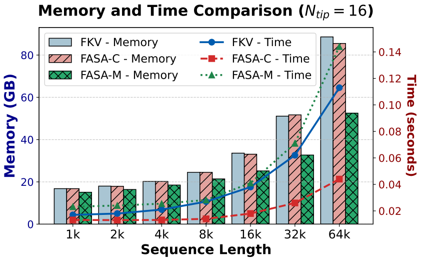


Figure 7: Memory vs. latency ($N_{tip} = 16$).

Efficiency Analysis. We assess the efficiency of our two FASA variants. FASA-M's memory savings are particularly pronounced in long sequences, as the KV cache's footprint grows to dominate and dwarf the static memory costs of model parameters and activations. While its CPU-GPU data transfer introduces a slight latency overhead, this can be effectively mitigated by prefetching techniques that asynchronously load the required KV pairs in advance. FASA-C, implemented with Triton (based on Ribar et al. (2024)), delivers substantial inference acceleration. The speedup effect intensifies with longer sequences, achieving up to a 2.56× with $N_{tip} = 16$ under 64K.

5.4 Ablation Studies

Robustness to Calibration Window $\mathcal{K}$. Our method exhibits remarkable robustness to the calibration window size, $\mathcal{K}$. Performance is largely insensitive to $\mathcal{K}$, with smaller $\mathcal{K}$ values often yielding slightly superior results (Table 5). This suggests that due to the inherent sparsity of attention, even a small calibration window provides a sufficiently robust signal to identify the dominant FCs.

Trade-off between N_{tip} and N_{fac}. The hyperparameters N_{tip} (token selection precision) and N_{fac} (retention budget) govern a trade-off between the fidelity of token identification and the volume of retained context. As depicted in Figure 6, optimal performance can be achieved either with high-precision selection (large N_{tip}) and a small budget, or a more lenient selection (small N_{tip}) compensated by a larger one. Empirically, on the TREC dataset, we found that using just 10 dominant FCs (15.6% of dimensions) with $N_{fac} = 500$ is sufficient to match the FKV's performance.

Impact of Offline Calibrated Data. As shown in Table 6, our method exhibits remarkable robustness to the choice of calibration data. The minimal performance variation across different calibration datasets, as quantified by a low Coefficient of Variation (CV), confirms that our FC detection mechanism is stable and not reliant on a specific calibration source.

Ablation Study on Data Size. As shown in Table 12, The robustness of our dominant FC identification is evident in the stable performance across all calibration set sizes. Crucially, this stability is achieved with as few as two QA pairs, demonstrating the high efficiency of FASA's offline calibration.

5.5 Generalization to Other Positional Encodings

Functional Sparsity on Other PEs **For ALiBi** (Baichuan-13B-Chat), as shown in Figure 8, the attention heads exhibit two patterns: one group shows the expected functional sparsity, while another shows extremely high contextual awareness across all dimensions. **This demonstrates that FASA is highly compatible with ALiBi models**; **For Partial-RoPE** (DeepSeek-V2-Lite-Chat), head dimension consists of both non-RoPE dimensions and RoPE frequency chunks. We computed CA scores for both parts and found a clear pattern that, **consistently aligns with our functional sparsity hypothesis** in Figure 9. Therefore, other PEs could also induce the functional sparsity.

Table 7: Performance on Partial-RoPE Models.

	Qasper	2Wiki	Multi	Passage_Re	Lcc	Samsum
FKV	33.18	19.83	47.27	49.00	63.40	34.04
FASA	33.46	20.25	46.50	48.50	62.49	32.53

Table 8: Performance on ALiBi Models.

	Qasper	Lsht	Dureader	Trec	Repobench
FKV	9.11	24.25	23.18	23.00	17.30
FASA	7.80	21.25	21.70	21.50	16.46

FASA Evaluation on Other PEs Ultimately, our work establishes FASA as a broadly applicable method, not confined to RoPE. This generalizability to diverse PE architectures is achieved at no significant performance cost, with results remaining on par with FKV.

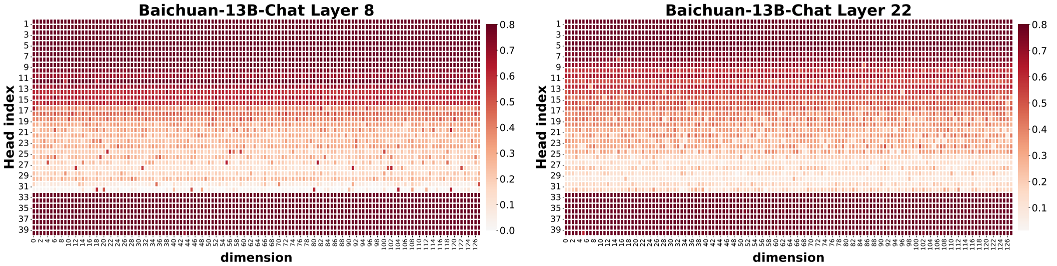


Figure 8: CA Scores Heatmaps of Baichuan-13B-Chat (ALiBi models).

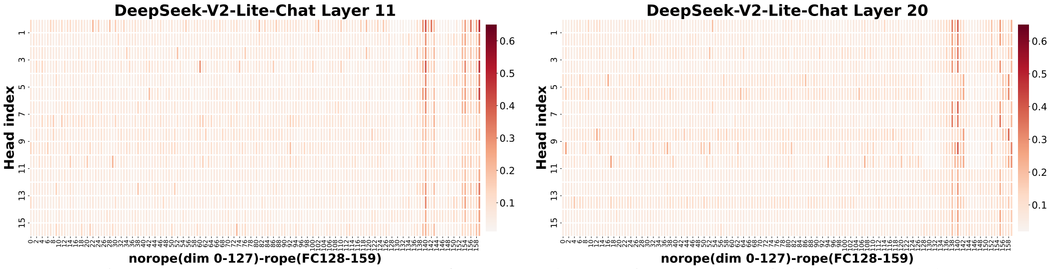


Figure 9: CA Scores Heatmaps of DeepSeek-V2-Lite-Cha (Partial-RoPE models).

6 Conclusion

In this work, we addressed the memory footprint and bandwidth introduced by the KV cache in LLMs. Firstly, we cover an intriguing phenomenon: the functional sparsity of FCs. A subset of dominant FCs could show high contextual awareness. Based on this discovery, we introduce FASA, a coarse-to-fine two-stage freamwork. The first stage utilizes the dominant FCs to perform dynamic, query-aware token selection without costly training. Then, the second stage perform focused and precise attention computation on this reduced subset. Our experiments indicate that FASA attains performance nearly on par with full KV even under constrained budgets. The memory- and speed-optimized variants of FASA offers a practical and effective solution for efficient long-context inference.

Ethics Statement

Our research is focused on enhancing the computational efficiency of Large Language Model (LLM) inference by optimizing KV cache management. The primary positive impact of our work, FASA, is to make large-scale models more accessible, affordable, and environmentally sustainable. By significantly reducing memory and computational overhead, our method can enable researchers and institutions with limited resources to develop and deploy powerful long-context models, thereby fostering broader innovation and democratization in the field of AI.

We acknowledge the dual-use nature of efficiency-enhancing technologies. While our goal is positive, lowering the barrier to running large models could inadvertently make it easier for malicious actors to deploy them for harmful purposes, such as generating misinformation or spam at scale. It is important to note, however, that our work is foundational and does not create new capabilities for generating harmful content; it merely optimizes the performance of existing models.

All experiments were conducted on publicly available benchmarks (LongBench, MATH, AIME) and open-source pre-trained models. We did not use any private, sensitive, or user-generated data. We recognize that the foundation models used in our evaluation may reflect and perpetuate societal biases present in their vast training corpora. Our method operates orthogonally to the challenge of model-level bias and does not address it directly, but we encourage users to be mindful of the inherent limitations of the models they deploy with our technique.

Reproducibility Statement

To ensure the reproducibility of our work, we provide a detailed account of all models, datasets, experimental setups, and evaluation protocols, all of which are publicly available. An overview of the experiments is provided in **Section 5.1**, with more comprehensive details described across several appendices. Specifically, the configurations for all baselines and the detailed hyperparameters for FASA are presented in **Appendix C.1**. The descriptions of all benchmarks and their corresponding evaluation protocols are detailed in **Appendix C.2** and **Appendix C.3**, respectively. Furthermore, the implementation and design choices for FASA are explained in **Appendix C.4**. Finally, the specific algorithms for FASA-M and other core functions are provided in **Appendix E.1** and **Appendix E.3**.

References

Eigen attention: Attention in low-rank space for kv cache compression, 2024. URL https://arxiv.org/abs/2408.05646.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=hmOwOZWzYE.

Yash Akhauri, Ahmed F AbouElhamayed, Yifei Gao, Chi-Chih Chang, Nilesh Jain, and Mohamed S. Abdelfattah. Tokenbutler: Token importance is predictable, 2025. URL https://arxiv.org/abs/2503.07518.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 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)*, pp. 3119–3137, 2024.

Federico Barbero, Alex Vitvitskyi, Christos Perivolaropoulos, Razvan Pascanu, and Petar Veličković. Round and round we go! what makes rotary positional encodings useful? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=GtvuNrk58a.

Payman Behnam, Yaosheng Fu, Ritchie Zhao, Po-An Tsai, Zhiding Yu, and Alexey Tumanov. Rocketkv: Accelerating long-context llm inference via two-stage kv cache compression, 2025. URL https://arxiv.org/abs/2502.14051.

Zefan Cai, Wen Xiao, Hanshi Sun, Cheng Luo, Yikai Zhang, Ke Wan, Yucheng Li, Yeyang Zhou, Li-Wen Chang, Jiuxiang Gu, et al. R-kv: Redundancy-aware kv cache compression for training-free reasoning models acceleration. *arXiv preprint arXiv:2505.24133*, 2025a.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling, 2025b. URL https://arxiv.org/abs/2406.02069.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Mohamed S. Abdelfattah, and Kai-Chiang Wu. Palu: KV-cache compression with low-rank projection. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LWMS4pk2vK.

Guoxuan Chen, Han Shi, Jiawei Li, Yihang Gao, Xiaozhe Ren, Yimeng Chen, Xin Jiang, Zhenguo Li, Weiyang Liu, and Chao Huang. Sepllm: Accelerate large language models by compressing one segment into one separator, 2025. URL https://arxiv.org/abs/2412.12094.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

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

Harry Dong, Xinyu Yang, Zhenyu Zhang, Zhangyang Wang, Yuejie Chi, and Beidi Chen. Get more with less: Synthesizing recurrence with kv cache compression for efficient llm inference, 2024. URL https://arxiv.org/abs/2402.09398.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive KV cache compression for LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uNrFpDPMyo.

Tanya Goyal and Greg Durrett. Evaluating factuality in generation with dependency-level entailment, 2020. URL https://arxiv.org/abs/2010.05478.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Monishwaran Maheswaran, Sebastian Zhao, June Paik, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. Squeezed attention: Accelerating long context length llm inference, 2025a. URL https://arxiv.org/abs/2411.09688.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. Kvquant: Towards 10 million context length llm inference with kv cache quantization, 2025b. URL https://arxiv.org/abs/2401.18079.

Dongsheng Jiang, Yuchen Liu, Songlin Liu, Jin'e Zhao, Hao Zhang, Zhen Gao, Xiaopeng Zhang, Jin Li, and Hongkai Xiong. From clip to dino: Visual encoders shout in multi-modal large language models. *arXiv preprint arXiv:2310.08825*, 2023.

Haoyang LI, Yiming Li, Anxin Tian, Tianhao Tang, Zhanchao Xu, Xuejia Chen, Nicole HU, Wei Dong, Li Qing, and Lei Chen. A survey on large language model acceleration based on KV cache management. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=z3JZzu9EA3.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024a.

Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. Minicache: KV cache compression in depth dimension for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=sgVOjDqUMT.

Di Liu, Meng Chen, Baotong Lu, Huiqiang Jiang, Zhenhua Han, Qianxi Zhang, Qi Chen, Chengruidong Zhang, Bailu Ding, Kai Zhang, Chen Chen, Fan Yang, Yuqing Yang, and Lili Qiu. Retrievalattention: Accelerating long-context llm inference via vector retrieval, 2024c. URL https://arxiv.org/abs/2409.10516.

Guangda Liu, Chengwei Li, Jieru Zhao, Chenqi Zhang, and Minyi Guo. Clusterkv: Manipulating llm kv cache in semantic space for recallable compression, 2025. URL https://arxiv.org/abs/2412.03213.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. Scissorhands: Exploiting the persistence of importance hypothesis for llm kv cache compression at test time, 2023. URL https://arxiv.org/abs/2305.17118.

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

MAA. American invitational mathematics examination - aime. In *American Invitational Mathematics Examination - AIME 2024*, February 2024. URL https://maa.org/math-competitions/american-invitational-mathematics-examination-aime.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Alexander Peysakhovich and Adam Lerer. Attention sorting combats recency bias in long context language models. *arXiv preprint arXiv:2310.01427*, 2023.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling, 2019. URL https://arxiv.org/abs/1911.05507.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019.

Luka Ribar, Ivan Chelombiev, Luke Hudlass-Galley, Charlie Blake, Carlo Luschi, and Douglas Orr. Sparq attention: Bandwidth-efficient LLM inference. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024. URL https://openreview.net/forum?id=Ue8EHzaFI4.

Prajwal Singhania, Siddharth Singh, Shwai He, Soheil Feizi, and Abhinav Bhatele. Loki: Low-rank keys for efficient sparse attention. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=raABeiV71j.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Hanshi Sun, Li-Wen Chang, Wenlei Bao, Size Zheng, Ningxin Zheng, Xin Liu, Harry Dong, Yuejie Chi, and Beidi Chen. Shadowkv: Kv cache in shadows for high-throughput long-context llm inference, 2025. URL https://arxiv.org/abs/2410.21465.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=F7aAhfitX6.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context llm inference. In *International Conference on Machine Learning*, pp. 47901–47911. PMLR, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Zhongwei Wan, Xinjian Wu, Yu Zhang, Yi Xin, Chaofan Tao, Zhihong Zhu, Xin Wang, Siqi Luo, Jing Xiong, Longyue Wang, and Mi Zhang. D_{2}O: Dynamic discriminative operations for efficient long-context inference of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HzBfoUdjHt.

Ao Wang, Hui Chen, Jiaxin Li, Jianchao Tan, Kefeng Zhang, Xunliang Cai, Zijia Lin, Jungong Han, and Guiguang Ding. Prefixkv: Adaptive prefix kv cache is what vision instruction-following models need for efficient generation, 2025a. URL https://arxiv.org/abs/2412.03409.

Zheng Wang, Boxiao Jin, Yuming Chang, Zhongzhi Yu, and Minjia Zhang. Model tells you where to merge: Adaptive KV cache merging for LLMs on long-context tasks, 2025b. URL https://openreview.net/forum?id=Q5VlpYRxGF.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Jian Tong, Haodong Duan, Qipeng Guo, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. VideoroPE: What makes for good video rotary position embedding? In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=tO7OVZkCo1.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. URL https://doi.org/10.48550/arXiv.2412.15115.

Qingyue Yang, Jie Wang, Xing Li, Zhihai Wang, Chen Chen, Lei Chen, Xianzhi Yu, Wulong Liu, Jianye Hao, Mingxuan Yuan, and Bin Li. Attentionpredictor: Temporal pattern matters for efficient llm inference, 2025. URL https://arxiv.org/abs/2502.04077.

Rongzhi Zhang, Kuan Wang, Liyuan Liu, Shuohang Wang, Hao Cheng, Chao Zhang, and yelong shen. LoRC: Low-rank compression for LLMs KV cache with a progressive compression strategy, 2025. URL https://openreview.net/forum?id=NI8AUSAc4i.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Re, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=RkRrPp7GKO.

A Rebuttal Section

A.1 Generalization on ALiBi and MLA

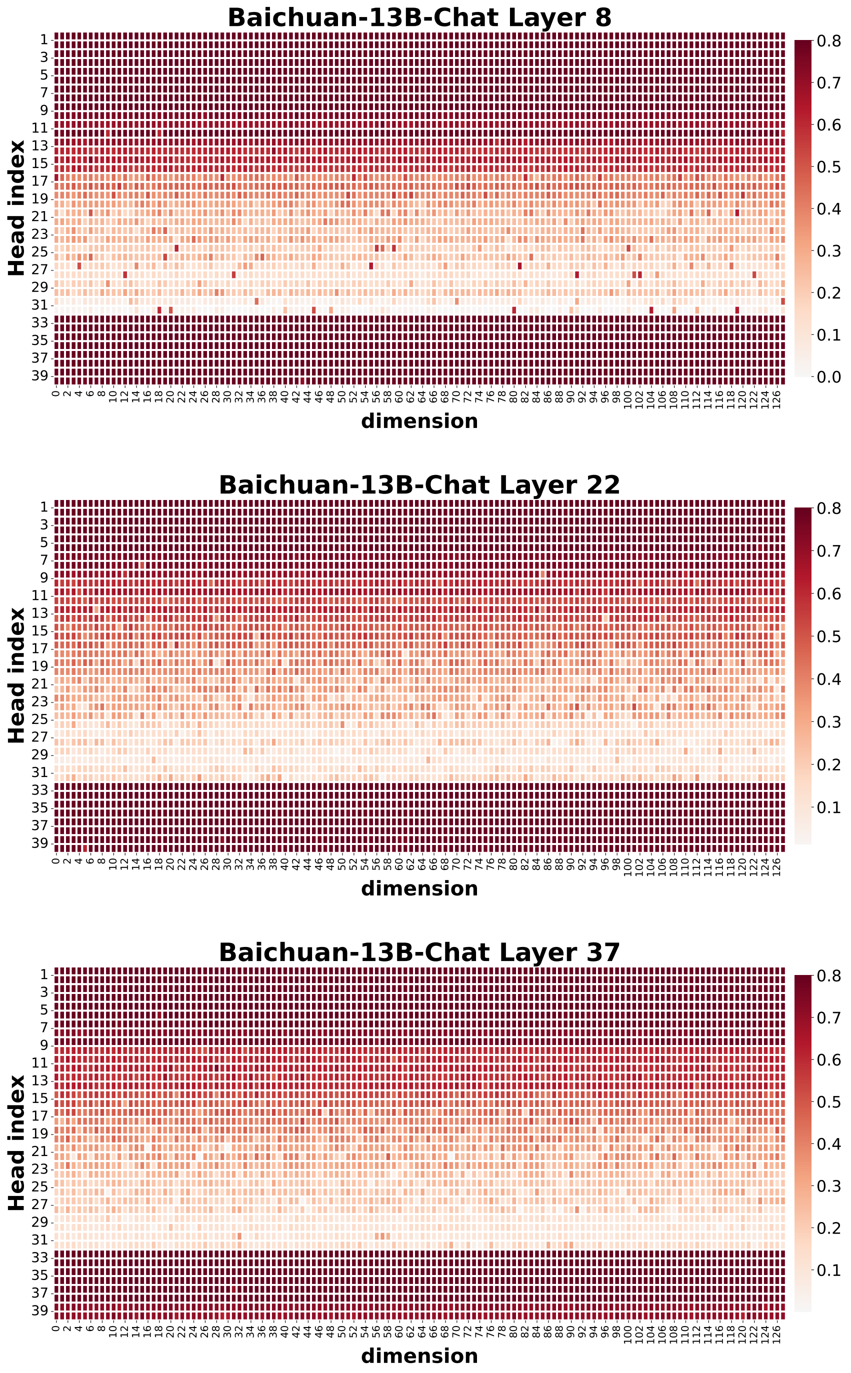


Figure 10: CA Scores Heatmaps of Baichuan-13B-Chat (ALiBi models).

We considered two prominent PE schemes: **ALiBi** (Attention with Linear Biases) and a **Partial-RoPE** hybrid, using the representative LLMs **Baichuan-13B-Chat** and **DeepSeek-V2-Lite-Chat**, respectively. We conducted experiments to test our functional sparsity hypothesis and evaluate FASA's performance on these models.

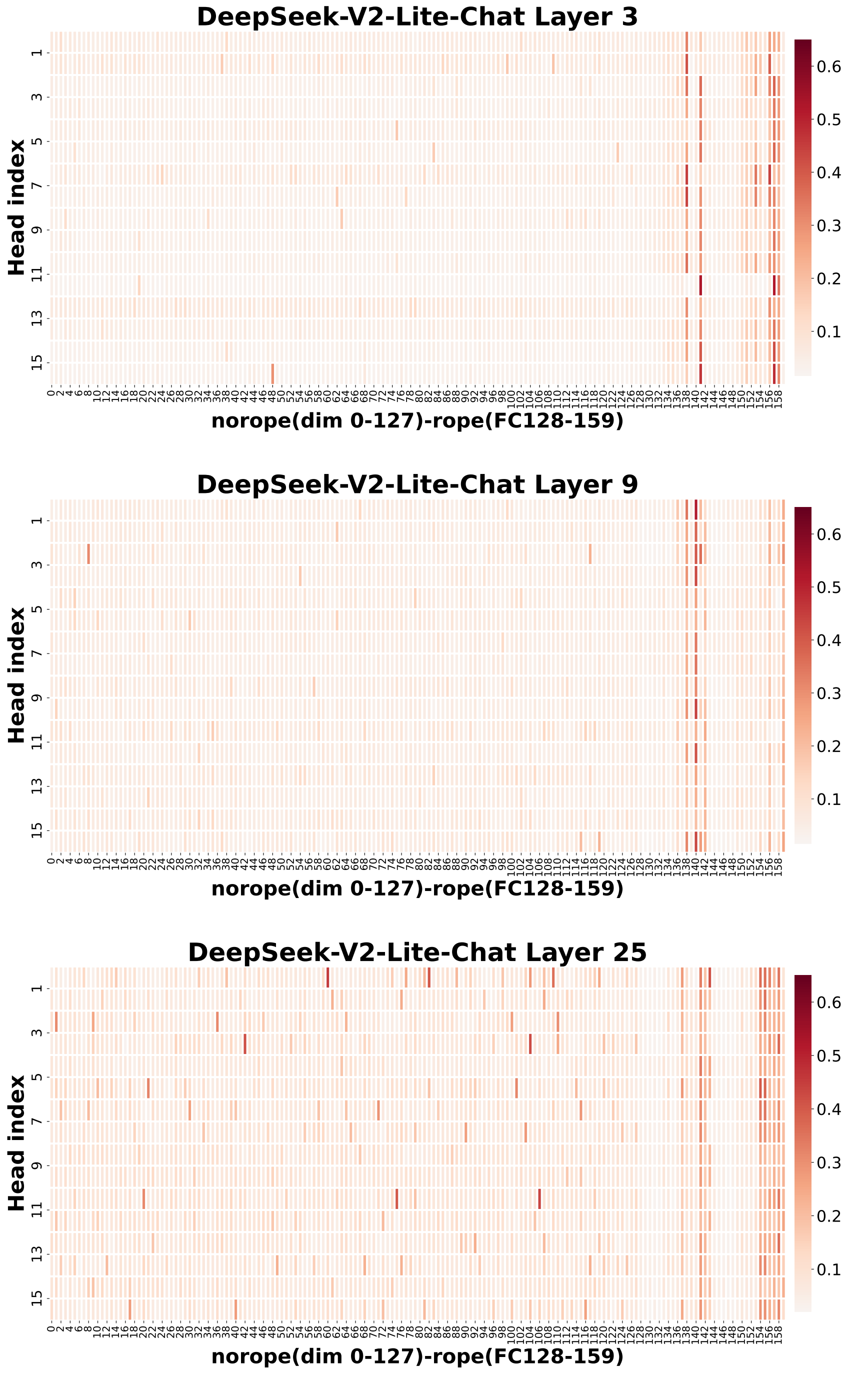


Figure 11: CA Scores Heatmaps of DeepSeek-V2-Lite-Chat (Partial-RoPE models).

a. Functional Sparsity on ALiBi and Partial-RoPE

- **Baichuan:** the attention heads exhibit two patterns: one group shows the expected functional sparsity, while another shows extremely high contextual awareness across all dimensions. **This demonstrates that FASA is highly compatible with ALiBi models**. (Figure 10)
- **DeepSeek-V2**: The head dimension consists of both non-RoPE dimensions and RoPE frequency chunks. We computed CA scores for both parts and found a clear pattern that, **consistently aligns with our functional sparsity hypothesis**. (Figure 11)

b. FASA Evaluation on Baichuan and DeepSeek-V2.

Table 9: FASA Evaluation on DeepSeek-V2-Lite-Chat ($N_{fac} = 256$).

Partial-RoPE	**Qasper**	**2Wikimqa**	**Multifieldqa**	**Passage_Re**	**Lcc**	**Samsum**
FKV	33.18	19.83	47.27	49.00	63.40	34.04
FASA	33.46	20.25	46.50	48.50	62.49	32.53

Table 10: FASA Evaluation on Baichuan-13B-Chat ($N_{fac} = 256$).

ALiBi	**Qasper**	**Lsht**	**Dureader**	**Trec**	**Lcc**	**Repobench**
FKV	9.11	24.25	23.18	23.00	14.29	17.30
FASA	7.80	21.25	21.70	21.50	13.62	16.46

Ultimately, our work establishes FASA as a broadly applicable method, not confined to RoPE. This generalizability to diverse PE architectures is achieved at no significant performance cost, with results remaining on par with FKV.

A.2 Generalization on Page-level Methods

The results of applying FASA to page-level methods are presented in Table 11, where page size n denotes the number of tokens per page. From these results, **we draw three key conclusions:**

- **Compatibility and Efficiency:** FASA is fully compatible with page-level selection and significantly enhances its computational efficiency.
- **Performance Maintenance:** When integrated into page-level methods, FASA maintains competitive performance across various page sizes, even while using only dominant FCs.
- **Superiority of Token-Level Granularity:** As a native token-level method, FASA substantially outperforms all page-level variants. Notably, this performance gap widens as the page size n increases.

Table 11: Performance comparison of FASA on page-level methods. The results demonstrate that FASA is applicable to page-level selection, enhancing efficiency while maintaining competitive performance across various page sizes.

Method	Qasper	Multifieldqa_en	Hotpotqa	2Wikimqa	Musique	Dureader	Avg.
Model: Mistral-7B-Instruct-v0.3							
FKV	41.60	52.90	49.40	39.50	29.10	30.96	40.58
FASA (ours)	41.48	53.81	49.22	40.01	28.80	32.00	40.89
Page-level Methods							
(page _size=8)	37.33	49.58	49.83	36.06	25.92	26.33	37.51
+ FASA	37.29	49.69	50.02	34.45	25.20	31.51	38.03
(page_size=16)	38.37	49.24	47.67	32.45	25.17	25.16	36.34
+ FASA	38.06	48.16	49.04	33.70	25.08	31.05	37.52
(page _size=32)	38.01	49.55	47.59	32.81	22.10	23.05	35.52
+ FASA	35.34	47.83	47.59	31.35	22.17	26.92	35.20
Model: Qwen2.5-7B-Instruct							
FKV	43.50	52.10	55.90	46.90	28.60	29.82	42.80
FASA (ours)	42.97	52.58	58.29	45.97	30.43	29.08	43.22
Page-level Methods							
(page _size=8)	42.42	51.58	56.56	46.64	29.46	23.40	41.68
+ FASA	41.58	52.31	56.96	46.42	27.60	30.28	42.53
(page _size=16)	42.05	51.78	56.36	45.54	28.08	22.82	41.11
+ FASA	41.46	50.84	55.41	45.14	27.04	27.63	41.25
(page _size=32)	41.65	51.89	56.19	46.33	28.10	22.34	41.08
+ FASA	40.20	50.41	54.42	43.38	26.92	27.54	40.48

A.3 Ablation Study on Data Size

Our ablation study on data size involves identifying dominant FCs using varying numbers of QA pairs. We then perform two analyses:

- Evaluate the performance of these FCs on **down-stream long-text tasks**.
- Measure **the percentage of overlap** between the sets of dominant FCs identified under each condition.

Table 12: Ablation study on data size used to identify dominant frequency chunks (Llama-3.2-3B-Ins).

Num. of QA	Narrativeqa	Qasper	Multifieldqa	Hotpotqa	2Wikimqa	Musique	Avg.
2	23.18	37.37	50.34	49.31	39.44	21.98	36.94
4	22.49	37.17	50.79	49.52	39.43	21.62	36.84
6	23.90	37.71	52.25	49.65	39.24	21.78	37.42
8	24.32	37.28	51.43	50.22	39.16	21.62	37.34
10	22.96	36.67	51.83	48.66	39.43	21.21	36.80

Conclusion (Downstream tasks): The robustness of our dominant FC identification is evident in the stable performance across all calibration set sizes. Crucially, this stability is achieved with as few as two QA pairs, demonstrating the high efficiency of FASA's offline calibration.

For a more direct analysis, we measure the overlap among the dominant FC sets identified with varying calibration data sizes.

Conclusion (Overlap Analysis): The robustness of the offline calibration process is confirmed by the high degree of overlap—consistently above 80%—among the dominant FC sets identified with different data sizes. This stability indicates that the identified FCs are not sensitive to the size of the calibration dataset.

Table 13: Overlap percentage (%) of dominant FCs identified using different numbers of QA pairs for calibration. The high degree of overlap demonstrates the robustness of the identification method.

Num. of QA	**2**	**4**	**6**	**8**	**10**
2	100.0	82.8	81.7	82.0	81.7
4		100.0	86.8	86.1	86.7
6			100.0	95.9	95.1
8				100.0	96.6
10					100.0

Summary: The results from downstream tasks and the overlap analysis demonstrate that our offline calibration process is both robust and efficient.

Ablation with N_{tip} on More Datasets Building upon the analysis for TREC and MATH presented in Figure 6, we now evaluate performance on four more diverse datasets to assess the generalizability of our approach across varying (N_{tip}) and constant $N_{fac} = 256$ in Table 14.

Table 14: Ablation study on the number of dominant FCs across more datasets under 256 token budget.

		Number of Dominant FCs					
Model	**calibrated dataset**	**8**	**10**	**12**	**14**	**16**	**FKV**
Mistral-7B-Instruct	Qasper	38.54	39.08	41.57	41.78	42.30	41.60
	Narrativeqa	26.26	28.58	28.67	29.26	29.90	29.10
	Dureader	26.65	28.50	29.72	29.73	30.21	30.96
	Gov_Report	29.82	31.61	33.25	33.96	34.00	34.80
	Avg.	**30.32**	**31.94**	**33.30**	**33.68**	**34.10**	**34.12**
Qwen2.5-14B-Instruct	Qasper	41.17	45.14	44.98	45.29	45.49	45.34
	Narrativeqa	25.91	26.38	27.87	28.50	30.40	29.71
	Dureader	23.72	23.71	25.42	26.68	28.92	29.32
	Gov_Report	24.41	26.56	27.24	28.62	29.51	29.71
	Avg.	**28.80**	**30.45**	**31.38**	**32.27**	**33.08**	**33.52**

Benefits of prefetching techniques We sincerely thank the reviewer for this insightful suggestion. You are correct to point out the trade-off: while FASA-M achieves significant memory savings, the CPU-to-GPU data transfer for a small fraction of tokens introduces latency. This effect is quantified in the third row of our results, which shows an increase in decoding time without prefetching. This is precisely why we introduced the prefetching technique, to counteract this specific overhead. As our results demonstrate, prefetching successfully mitigates this latency, bringing the overall decoding time to a level comparable to the baseline.

Table 15: Comparision results with and without prefetching techniques.

	1k	**2k**	**4k**	**8k**	**16k**	**32k**	**64k**
base	0.018	0.019	0.023	0.027	0.038	0.062	0.113
W. prefetch	0.021	0.024	0.026	0.031	0.046	0.086	0.154
Wo. prefetch	0.028	0.038	0.049	0.066	0.128	0.185	0.339

A.4 Quantitative Results on Sparsity &Task-Invariance

Sparsity: We quantitatively analyzed the proportion of dominant FCs (defined as CA > 0.4). We found they account for less than 1% of all FCs, while non-dominant FCs with low CA scores comprise approximately 90% or more. This empirically validates our claim of functional sparsity.

Universality: This sparsity pattern holds universally. We confirmed its existence across different architectures (Llama, Qwen, Mistral, R1 models) and scales (3B to 32B), which strongly supports our universality claim.

Table 16: The ratio of dominant FCs and non-dominant FCs.

Type of FC	Dominant FCs (%)	Non-Dom FCs(%)
Model	**CA scores $>$ 0.4**	**CA score $<$ 0.15**
Llama-3.2-3B	0.54	89.6
Meta-Llama-3.1-8B	0.68	89.6
Mistral-7B-v0.3	0.68	92.7
Qwen2.5-7B	0.17	95.5
Qwen2.5-14B	0.27	94.7
Qwen2.5-14B-1M	0.65	90.5
Qwen2.5-32B-Instruct	0.52	91.2
R1-Distill-Llama-8B	0.79	89.5
R1-Distill-Qwen-14B	0.76	90.2
R1-Distill-Qwen-32B	0.67	90.9

Table 17: Cross-task overlap matrix of dominant FCs (%). Each sub-table shows the percentage of intersection between dominant FCs identified on a "row" dataset and a "column" dataset.

Model	**Overlap of dom-FCs**	**Qasper**	**Gov_Report**	**Musique**	**Narrativeqa**	**2Wikimqa**	**Avg.**
Llama-3.2-3B	**Qasper**	100.00	75.90	82.30	70.50	83.20	82.38
	Gov_Report	75.90	100.00	82.10	70.80	81.90	82.14
	Musique	82.30	82.10	100.00	73.60	96.50	86.90
	Narrativeqa	70.50	70.80	73.60	100.00	73.10	77.60
	2Wikimqa	83.20	81.90	96.50	73.10	100.00	86.94
Mistral-7B	**Qasper**	100.00	71.10	77.10	67.30	77.00	78.50
	Gov_report	71.10	100.00	79.40	65.50	78.90	78.98
	Musique	77.10	79.40	100.00	67.80	97.90	84.44
	Narrativeqa	67.30	65.50	67.80	100.00	67.30	73.58
	2Wikimqa	77.00	78.90	97.90	67.30	100.00	84.22
Qwen2.5-7B	**Qasper**	100.00	70.60	80.90	68.70	81.30	80.30
	Gov_Report	70.60	100.00	79.40	68.20	78.70	79.38
	Musique	80.90	79.40	100.00	71.70	96.60	85.72
	Narrativeqa	68.70	68.20	71.70	100.00	71.10	75.94
	2Wikimqa	81.30	78.70	96.60	71.10	100.00	85.54
Qwen2.5-14B	**Qasper**	100.00	69.20	84.30	71.80	84.50	81.96
	Gov_Report	69.20	100.00	75.00	67.60	74.80	77.32
	Musique	84.30	75.00	100.00	74.30	98.40	86.40
	Narrativeqa	71.80	67.60	74.30	100.00	73.90	77.52
	2Wikimqa	84.50	74.80	98.40	73.90	100.00	86.32

Task-Invariance: To provide direct evidence of task-invariance, we measured the overlap between sets of dominant FCs identified using different calibration datasets. Our analysis reveals a remarkably high degree of overlap, which consistently exceeds 70% across all tested models and tasks.

This finding offers compelling evidence for the task-agnostic nature of these dominant FCs. The effect is particularly pronounced in the Llama model, where the overlap between the sets identified by the Musique and 2WikiMQA datasets surpasses 90%. Such high consistency strongly indicates that the set of dominant FCs is not determined by the calibration task, but is rather an intrinsic, emergent property of the model's fundamental architecture. This conclusion is generalizable, as the pattern holds true across models of varying scales and designs.

Long-CoT Performance with Long-context Calibration we evaluated the R1 models on MATH tasks using FCs calibrated on long-context tasks (Table 18 and 19). Our findings are twofold:

(1) Models calibrated on a long-context task achieve performance on the MATH that is highly comparable to counterparts calibrated directly on MATH. For instance, R1-Qwen-14B and R1-Qwen-32B achieve 91.0% accuracy on MATH (1000-token budget) when calibrated with Qasper. Furthermore, R1-Distill-Llama-8B consistently delivers performance comparable to the FKV baseline on both MATH and AIME, regardless of the calibration datasets. These results provide strong evidence for the robustness and task-agnostic nature of the dominant FC identification.

(2) We did, however, observe a minor exception. For the R1-Distill-Qwen-32B model on MATH, when the token budget is restricted to 300 tokens, the version calibrated on Qasper performs slightly below the version calibrated on MATH itself. This finding does not contradict our main conclusion. We hypothesize that the activation patterns of very short outputs diverge from the activation distribution of longer outputs, which are more representative of the R1 model's intrinsic dynamics, explaining the slight performance difference.

Table 18: Performance on the MATH dataset using FCs calibrated on different datasets.

		MATH (Token Budget)				
Model	**Calibration**	**300**	**500**	**700**	**1000**	**AVG**
R1-Distill-Qwen-14B	FKV	92.4	92.4	92.4	92.4	92.4
	MATH	86.6	88.8	90.2	91.2	89.2
	Qasper	87.2	89.2	91.0	91.0	89.6
R1-Distill-Qwen-32B	FKV	92.6	92.6	92.6	92.6	92.6
	MATH	86.4	90.2	90.2	91.2	89.5
	Qasper	79.8	84.8	86.6	90.6	85.5

Table 19: Performance on MATH and AIME datasets using diverse calibration datasets for R1-Distill-Llama-8B.

		MATH (Token Budget)			
Model	**Calibration**	**300**	**500**	**700**	**1000**
R1-Distill-Llama-8B	FKV	72.40	72.40	72.40	72.40
	Math	62.20	68.80	69.40	71.80
	AIME	63.20	67.60	71.80	72.00
	Qasper	57.10	64.60	68.40	71.80
	Gov_Report	58.60	60.40	70.40	71.60
	2Wikimqa	58.80	62.20	68.60	69.80

		AIME 24 (Token Budget)				
Model	**Calibration**	**500**	**1000**	**1500**	**2000**	**2500**
R1-Distill-Llama-8B	FKV	43.90	43.90	43.90	43.90	43.90
	Math	20.60	34.40	40.20	35.80	38.00
	Qasper	18.80	33.30	36.76	37.68	41.34

A.5 Distribution of CA Scores

Distribution of CA Scores We conducted an additional analysis specifically designed to examine the predicting accuracy of dominant and non-dominant FCs on tokens of varying attention magnitudes (i.e., importance levels).

Table 20: Prediction accuracy in each attention scale ranges.

		Prediction accuracy across varying attention scale ranges				
Model	**Type of FCs**	**Top 20%**	**Top 20-40%**	**Top 40-60%**	**Top 60-80%**	**Top 80-100%**
Llama-3.2-3B-Instruct	dom	82.4*	79.1	72.1	59.2	44.9
	non-dom	4.6	5.3	5.3	5.4	5.4
Mistral-7B-Instruct-v0.3	dom	81.1	80.7	78.7	72.5	56.4
	non-dom	3.6	4.2	4.9	4.4	4.5
Qwen2.5-7B-Instruct	dom	81.9	82.4	76.9	63.7	49.3
	non-dom	6.1	5.7	5.4	5.6	5.5
Qwen2.5-14B-Instruct	dom	74.3	66.4	56.6	44.9	34.7
	non-dom	4.1	4.6	4.5	4.9	4.9

PS: 82.4 means dominant FCs successfully predicts 82.4% tokens in Top-20%.

Predicting Performance Across Attention Score Ranges: We selected the top 256 tokens with the highest attention scores at each generation step and divided them into five token ranges based on attention magnitude: Top 0–20%, Top 20–40%, Top 40–60%, Top 60–80%, and Top 80–100%. For each token range, we calculated the proportion of tokens accurately predicted by dominant FCs/non-dominant FCs (where we defined CA scores above 0.6 as dominant FC and those below 0.15 as non-dominant FC).

- **Dominant FCs identify the most influential tokens:** The performance of dominant FCs is heavily concentrated in the token ranges with the highest attention scores. For instance, in the Llama-3.2-3B model, dominant FCs account for 82.4% of the prediction performance within the top 20% most important tokens. This contribution progressively declines as token importance decreases. This strongly indicates that **dominant FCs not only capture the overall relative ranking of token importance** (as reflected by high CA scores) but **also accurately capture the performance magnitude of the most influential tokens.**
- **Non-dominant FCs consistently show extremely low accuracy.** This finding directly counters the possibility that "some non-dominant FCs may simply rank poorly but still attend to the most important tokens." Experiments indicate that non-dominant FCs perform poorly in capturing both the influence and ranking of token importance.

Conclusion: We deeply appreciate your insightful suggestions. By presenting the distribution of CA scores across token ranges with varying attention magnitudes, this analysis further **substantiates the ability of dominant FCs to effectively capture both the relative ranking and true impact of context tokens.**

A.6 Cumulative Distribution Function (CDF) of CA scores

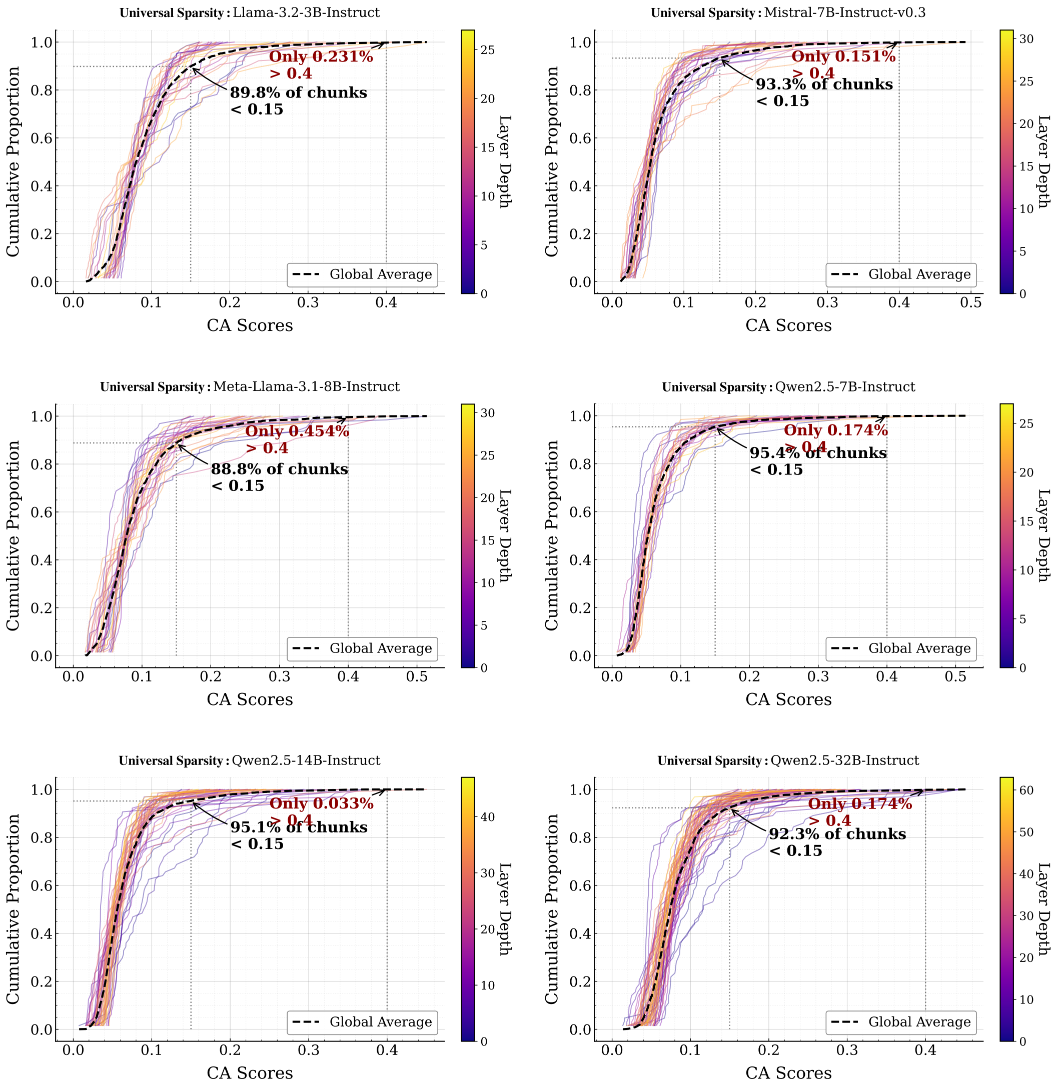


Figure 12: CDF figures of CA scores across all model layers and models.

Solid Evidence of Functional Sparsity To further substantiate the universality of functional sparsity, we analyze the CDF of CA scores across all layers of six evaluated LLMs (see Figure 12). As illustrated, the distribution is heavily skewed towards zero, demonstrating that most frequency chunks have low contextual awareness. Specifically, over 90% of chunks typically have scores below 0.15, while high-scoring FCs are exceptionally rare, with consistently less than 0.5% of FCs exhibiting strong contextual awareness (CA > 0.4). **These CDF plots confirm that functional sparsity is not an artifact of specific layers but a fundamental and universal property of the model architectures.**

B Investigation Results of Dominant Frequency Chunks

B.1 Further Generalization on Model Scales and Architechtures

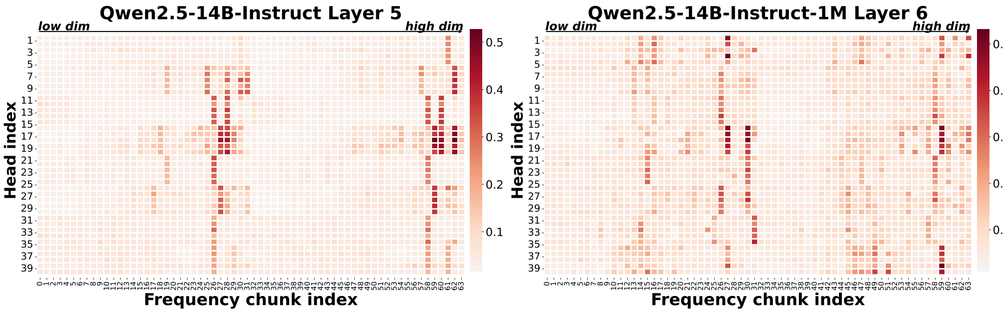


Figure 13: Functional sparsity is maintained on Qwen2.5 series models (Yang et al., 2024). Heatmaps visualize the Mean Contextual Agreement ($\overline{\text{CA}}_{K=256}$) for each Frequency Chunk (FC, x-axis) across all attention heads (y-axis) in a representative layer. We compare the standard **Qwen2.5-14B-Instruct** model (left) with its long-context variant, **Qwen2.5-14B-Instruct-1M** (right), both calibrated on the Qasper dataset. The remarkable similarity between the two heatmaps demonstrates that the functional sparsity of FCs is a robust property, consistently maintained even after long-context fine-tuning.

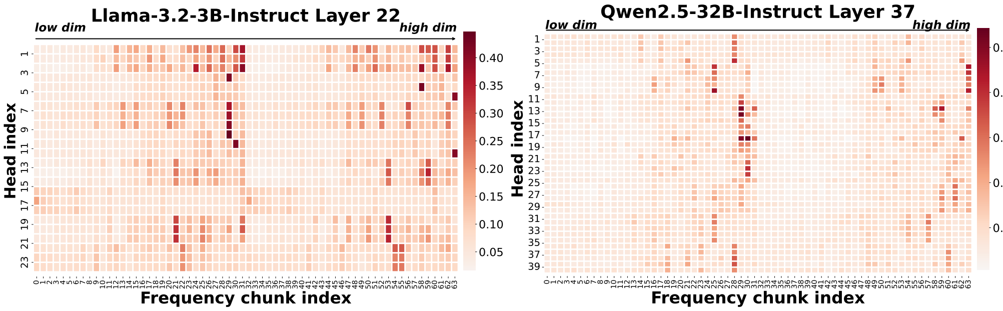


Figure 14: Functional sparsity persists across model scales. Heatmaps show the Mean Contextual Agreement ($\overline{\text{CA}}_{K=256}$) for increasing scale (3B and 32B). The remarkable stability of the dominant FC patterns (bright vertical columns) across these scales demonstrates that functional sparsity is a fundamental and scalable characteristic of RoPE.

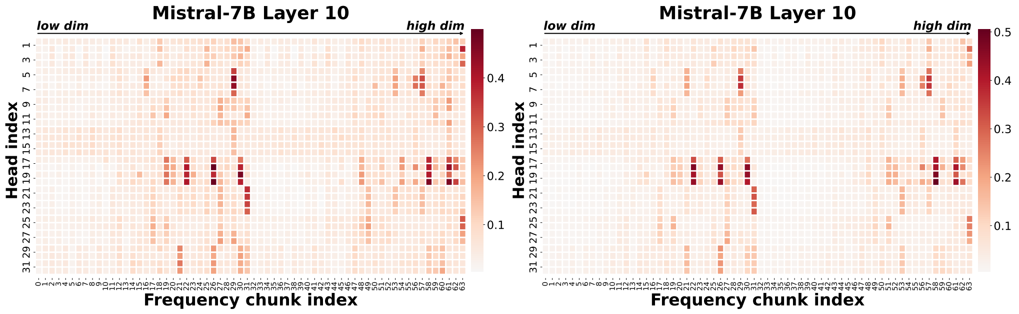
Conclusions: Our cross-architectural (Figure 13) and cross-scale (Figure 14) analysis reveals a striking finding: the functional sparsity of FCs is a universal and stable property. This powerful evidence suggests that the observed functional hierarchy is not an emergent artifact of a specific model's training dynamics or size, but rather an intrinsic characteristic deeply embedded within the RoPE mechanism itself. The roles of different frequencies appear to be fundamental and pre-determined, providing a robust and predictable foundation for developing model-agnostic efficiency optimizations.

B.2 Task-Invariance Property of Functional Sparsity

We find that the saliency of dominant FCs is largely task-agnostic. This property is evidenced by the strong alignment between saliency maps generated for distinct downstream tasks, as shown in Figure 15. Despite the functional differences between question answering (left) and summarization (right), the resulting importance rankings are highly consistent. This indicates that these FCs perform a fundamental role inherent to the model's architecture, rather than one adapted for a specific task.

B.3 More Analysis Results

Functional Sparsity across Layers. While the principle of functional sparsity is universal, the specific set of dominant FCs is far from static in Figure 16; instead, it exhibits a high degree of



(a) Qasper

(b) GovReport

Figure 15: Heatmaps of agreement score ($\overline{\text{CA}}$, $K = 256$) across attention heads for the Qasper (Left) and GovReport (Right) from LongBench-V1 (Bai et al. (2024)) on Mistral-7B-Instruct-v0.3.

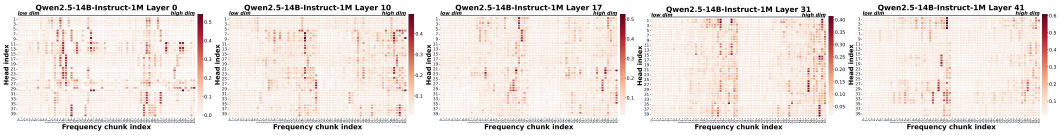


Figure 16: Heatmaps of agreement score ($\overline{\text{CA}}$, $K = 256$) across different layers.

specialization across both model depth and individual attention heads. This dynamic behavior reveals a sophisticated division of labor within the transformer architecture.

C Experiments Details

C.1 Experiment Configurations.

Baseline Configurations. As FASA is designed to optimize the decode phase, we forgo any KV cache optimizations during prefilling for all methods under evaluation. This experimental design isolates the performance impact of decode-stage acceleration, ensuring that our comparisons are direct and fair. For all baselines, we adopted configurations that are either standard in their original papers or represent a fair and strong setup for comparison.

- **Oracle**: serves as an oracle baseline to demonstrate the upper-bound performance of Top-k sparse attention. This method operates under the ideal assumption that the k most important KV tokens for each query can be identified perfectly and at no computational cost. Consequently, a given token budget directly corresponds to this optimal Top-k set.
- **Stream** (Xiao et al., 2024): This method is based on the "attention sink" phenomenon, preserving a fixed number of initial tokens and a sliding window of recent tokens. Following its standard setup, we set the initial "start_size" to 8 and the "recent_size" to "budget - 8".
- **SnapKV** (Li et al., 2024): SnapKV estimates token importance based on accumulated attention scores within a observation window during prefilling. We adopted its "maxpool" strategy with a window size of 32 and a kernel size of 7. As its original design performs a one-time filtering, it is not directly suited for long-generation tasks. We therefore adapted it, following the methodology in (Cai et al., 2025a), by re-applying the filtering mechanism every n generated tokens.
- **Quest** (Tang et al., 2024): Quest organizes the KV cache into pages and retrieves them based on a coarse-grained query-page similarity. We set the page size to 16, a value reported as near-optimal, to balance the trade-off between retrieval granularity and overhead.
- **RKV** (Cai et al., 2025a): RKV is a state-of-the-art method for reasoning tasks that also employs a retrieval mechanism. We set its core hyperparameter λ, which balances between recent and important tokens, to 0.1 as recommended for optimal performance.

FASA Configurations. Our configuration for FASA is designed for both effectiveness and practical efficiency. Unless otherwise specified, the following setup was used across all experiments.

- **Dominant FC Identification:** A core principle of FASA is that the set of dominant FCs is a universal, task-agnostic property of the model architecture itself. Consequently, these indices ($\mathcal{I}_{dom}$) can be determined via a highly efficient, one-time offline calibration. For our **LongBench** experiments, this calibration was performed on just a single data sample from the Qasper dataset. We found this minimal setup to be remarkably robust, as the generated response provides sufficient signal to identify the dominant FCs. The universality of these calibrated indices is empirically validated by FASA's strong performance across diverse tasks, from summarization to code completion. For **Long-CoT reasoning**, a similar single-instance calibration was performed on a question from the MATH500 dataset.
- **Hyperparameter Settings:** For architectural simplicity and to maximize computational parallelism, we employ a uniform configuration across all heads and layers. The number of dominant FCs to retain, denoted as N_{tip}, was consistently set to 16. This choice represents a balance between preserving sufficient contextual information and maximizing computational.
- **Task Configurations:** We configured the maximum sequence length to 32k for the AIME24 benchmark, reflecting its higher reasoning complexity, and to 16k for MATH500. For the LongBench benchmark, we set the maximum prompt length to 127.5k for Llama3/Qwen2.5 series models and 31.5k for Mistral-7B-Instruct-v0.2.

C.2 Benchmark Details

LongBench (Bai et al., 2024) is a comprehensive, multi-task benchmark designed to evaluate the long-context understanding capabilities of Large Language Models. It comprises a diverse set of tasks, including single-document QA, multi-document QA, summarization, few-shot learning, synthetic tasks, and code completion. In our experiments, we report the average performance across all relevant tasks to provide a holistic measure of a model's ability to process and reason over extended contexts, with sequence lengths ranging from 4K to over 100K tokens.

MATH500 (Hendrycks et al., 2021) is a challenging benchmark for evaluating mathematical reasoning. It consists of 12,500 problems sourced from high school math competitions, spanning subjects like Algebra, Geometry, Number Theory, and Precalculus. Each problem is accompanied by a step-by-step solution, making it highly suitable for assessing CoT reasoning capabilities. We utilize the MATH500 subset for our long-CoT generation experiments, where models must produce detailed reasoning chains to arrive at the final answer.

AIME (MAA, 2024) represents a significant step-up in reasoning complexity compared to the MATH dataset. It consists of problems from the AIME competition, which are known for their non-routine, multi-step solutions requiring deep mathematical insight and creativity. These problems serve as a stress test for a model's most advanced reasoning and long-chain generation abilities. Following standard practice, we evaluate performance using the pass@k metric, specifically reporting pass@1 based on 16 generated responses per question.

C4(Raffel et al., 2019) is a massive, general-domain English text dataset derived from the Common Crawl web scrape. The "clean" version is created by applying a series of heuristics to filter out boilerplate content, code, and offensive language, resulting in a high-quality, natural language corpus.

PG19 (Rae et al., 2019) is a long-form text dataset derived from books in the Project Gutenberg library. It is specifically curated for evaluating long-range sequence modeling. Each example in the dataset is a full book text, making it an ideal benchmark for assessing a model's ability to handle and maintain coherence over very long dependencies, often exceeding the context windows of LLMs.

WikiText(Merity et al., 2017) is a large-scale language modeling corpus sourced from high-quality "Good" and "Featured" articles on Wikipedia. Unlike raw web text, WikiText is well-formatted, grammatically correct, and retains its original punctuation and case. It is split into training, validation, and test sets at the article level.

C.3 Evaluation Protocols

To provide a comprehensive and rigorous assessment of model performance, we employ a set of standard metrics tailored to each evaluation paradigm.

Long-Context Understanding (LongBench). For the diverse tasks within the LongBench benchmark (Bai et al., 2024), we follow its official evaluation protocol. Specifically, we use:

- **f1 score** for question-answering tasks.
- **rouge_score** for summarization tasks.
- **code_sim_score** for code completion tasks.

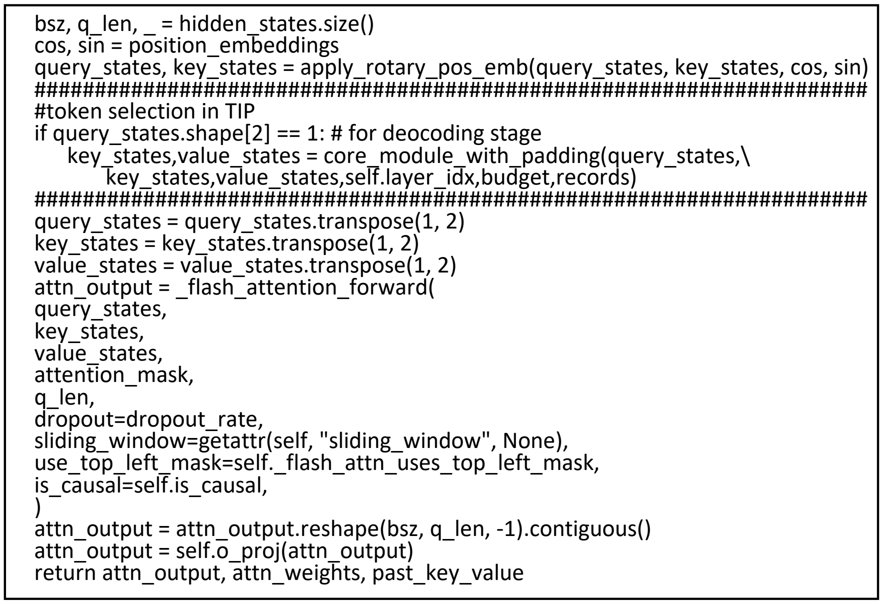
The final reported score for LongBench is the average performance across all constituent tasks.

Long-Sequence Modeling. To evaluate a model's ability to maintain generative fidelity over long dependencies, we use perplexity (PPL). Perplexity measures how well a probability model predicts a sample. For a sequence of tokens $W = (w_1, w_2, \ldots, w_N)$, PPL is defined as the exponential of the average negative log-likelihood in Equation 9. A lower PPL indicates a better model, as it signifies higher confidence and accuracy in predicting the next token.

$$\mathrm{PPL}(W) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i|w_{<i})\right) \tag{9}$$

Long CoT Reasoning. For complex mathematical reasoning tasks such as MATH500 and AIME2024, we evaluate the model's performance in a long-generation setting. This paradigm is distinct from conventional long-context understanding tasks. Instead of processing a long static input, the model must maintain logical coherence and track thought traces across an extended, auto-regressive generation process to produce the correct final answer. Performance is reported as pass@1.

- For MATH500, we report pass@1, where a single generation is sampled for each problem.
- For AIME2024, which features more challenging problems, we also report pass@1, but the result is determined by checking if at least one correct answer exists within $k = 16$ independent generations for each question. This sampling strategy is standard for estimating performance on complex reasoning benchmarks.



```
bsz, q_len, _ = hidden_states.size()
cos, sin = position_embeddings
query_states, key_states = apply_rotary_pos_emb(query_states, key_states, cos, sin)
######################################################################
#token selection in TIP
if query_states.shape[2] == 1: # for deocoding stage
    key_states,value_states = core_module_with_padding(query_states,\
        key_states,value_states,self.layer_idx,budget,records)
######################################################################
query_states = query_states.transpose(1, 2)
key_states = key_states.transpose(1, 2)
value_states = value_states.transpose(1, 2)
attn_output = _flash_attention_forward(
query_states,
key_states,
value_states,
attention_mask,
q_len,
dropout=dropout_rate,
sliding_window=getattr(self, "sliding_window", None),
use_top_left_mask=self._flash_attn_uses_top_left_mask,
is_causal=self.is_causal,
)
attn_output = attn_output.reshape(bsz, q_len, -1).contiguous()
attn_output = self.o_proj(attn_output)
return attn_output, attn_weights, past_key_value
```

Figure 17: The FASA Pipeline: An Efficient, FlashAttention-Compatible Approach. The algorithm details our two-stage process. A key design feature is that the FAC stage seamlessly integrates with the standard FlashAttention API, leveraging its performance while enabling sparse computation.

C.4 Implement Details

Implementation Details Our implementation of FASA is built upon the HuggingFace Transformers library (Wolf et al., 2020). We employ a non-invasive monkey patching approach to integrate our logic. Specifically, we intercept the forward pass of the FlashAttention2 class within the model's modeling.py file. The core of our method resides in two components. First, leveraging the universal nature of dominant FCs, their pre-computed indices are stored in a globally accessible dictionary, shared across all layers and heads. Second, the Token Importance Prediction (TIP) logic, which performs the critical token selection, is encapsulated within our core_module_with_padding function. A key advantage of our design is its simplicity and minimal intrusion. The integration requires inserting just a single line of code, the token selection logic, into the original attention function, making FASA easy to deploy and adapt. This minimal intrusion makes FASA highly portable and easy to adapt. The corresponding pseudocode is provided in Figure 17.

D Additional Experimental Results

D.1 Performance Analysis on different budgets

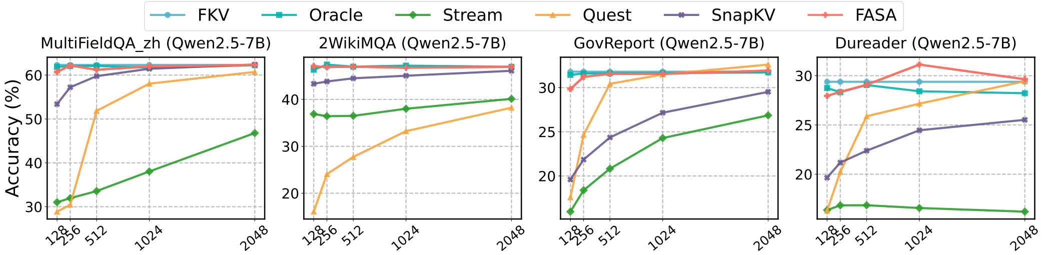


Figure 18: FASA on Qwen2.5-7B-Instruct under various token budgets ($N_{\text{tip}} = 16$).

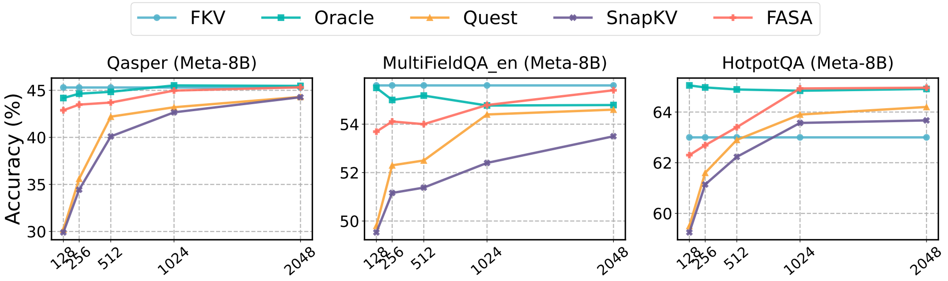


Figure 19: FASA on Meta-3.1-Llama-8B-Instruct under various token budgets ($N_{\text{tip}} = 16$).

Comparison with Low-Rank Methods A closely related work to FASA is SparQ (Ribar et al., 2024), which also performs a form of dimension selection. SparQ operates on the heuristic that high-magnitude dimensions in a query vector are the most indicative of importance, and thus selects corresponding key dimensions as a proxy for token prediction. However, as our experiments in Figure 20 demonstrate, this heuristic proves to be a poor substitute for true contextual awareness. Under a constrained budget of 256 tokens, SparQ's performance collapses, indicating its inability to reliably identify critical tokens based solely on query magnitudes. Furthermore, from an efficiency standpoint, SparQ incurs significant overhead as it must re-evaluate high-magnitude dimensions for every new query. In stark contrast, FASA leverages a one-time, offline calibration, making its per-token inference cost substantially lower.

E Discussion on FASA

E.1 Variants of FASA

FASA-M (Memory-Optimized) The memory-optimized variant, FASA-M, is specifically engineered for scenarios with constrained GPU memory, such as consumer-grade hardware. As detailed in Algorithm 2, its core strategy is to minimize the on-GPU memory footprint by strategically keeping only the most essential data on the GPU.

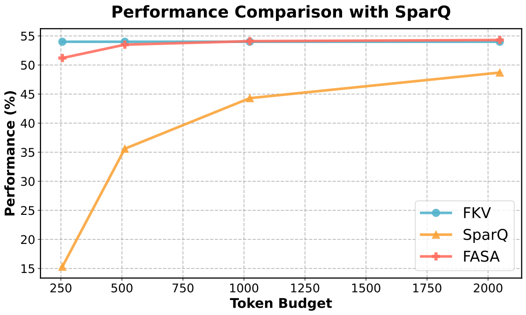


Figure 20: Comparision with SparQ on LongBench.

Specifically, only the dominant parts of the Key cache (C_{key}^{dom}), which are required for the initial token importance prediction, are retained in GPU memory. The non-dominant parts of the Key cache (C_{key}^{nondom}) and the entire Value cache (C_{val}) are offloaded to and managed in the much larger CPU memory. During the Focused Attention Computation (FAC) stage, once the critical token indices ($\mathcal{T}_t$) are identified, only the small, required subsets of the non-dominant key and value caches are transferred from the CPU to the GPU for the final attention calculation. This "just-in-time" data transfer ensures that the GPU memory is primarily occupied by the most critical components, leading to substantial memory savings.

Memory Footprint Analysis The GPU memory footprint of the KV cache in FASA-M can be formulated as follows. Let L be the total sequence length, b the token budget, d the model's hidden dimension, and N_{layers} the number of layers. Let d_{dom} be the dimension of the dominant FCs and d_{nondom} be the dimension of the non-dominant FCs ($d = d_{dom} + d_{nondom}$). The memory occupied by the KV cache on the GPU is:

$$\text{Mem}_{\text{GPU}} \approx N_{layers} \times \left(\underbrace{L \times d_{dom}}_{\text{Dominant Keys}} + \underbrace{b \times d_{nondom}}_{\text{Non-dominant Keys}} + \underbrace{b \times d}_{\text{Values}} \right) \times \text{bytes\_per\_param} \tag{10}$$

Compared to a full KV cache, which occupies $N_{layers} \times L \times 2d \times$ bytes_per_param, FASA-M significantly reduces the memory burden, especially when the non-dominant and value components constitute a large portion of the cache. For instance, if d_{dom} is 25% of d and the budget b is 10% of L, the memory savings can be substantial, approaching an $8\times$ reduction in typical configurations.

E.2 Design Choices

- **On the Role of FC-Scores: A Proxy for Ranking, Not a Substitute for Attention.** A crucial design principle we validated is that our FC-based scores ($\mathbf{S}_t^{l,h}$) are not calibrated to function as direct attention weights. Although they provide a remarkably accurate relative ranking of token importance, their direct substitution for attention probabilities leads to a catastrophic performance degradation. This reveals their fundamental role as a selector—a mechanism to identify salient tokens rather than an approximator of the final attention distribution.
- **On the Indivisibility of Frequency Chunks.** We investigated whether individual dimensions could serve as selection units, and the answer is a definitive no. A pipeline based on selecting "dominant dimensions" suffers a catastrophic performance degradation. This empirically validates that the Frequency Chunk (FC) is an indivisible functional unit for this process. This principle is not coincidental but is a direct corollary of RoPE's core mechanism, which encodes position by applying rotations to coupled pairs of dimensions. Disrupting these pairs severs the positional encoding, leading to model failure.

In summary, these two findings underscore two core design principles of FASA. First, an efficient proxy for token importance does not necessarily serve as a valid substitute for attention weights. Second, any optimization for RoPE-based models must respect the inherent coupling of dimension pairs, treating the Frequency Chunk as an indivisible functional unit.

E.3 Algorithm on FASA

See the algorithm of offline calibration in Algorithm 1; see the algorithm of FASA-M in Algorithm 2.

Algorithm 1: Offline Calibration for Dominant FCs

```
Input: A calibration dataset $\Omega$; number of dominant FCs to select $k$.
Output: The set of dominant FC indices, $\mathcal{I}_{dom}$.
// Stage 1: Collect Contextual Agreement (CA) scores
Initialize an empty map $M$ to store CA scores for each $(l, h, i)$ triplet
foreach example in $\Omega$ do
    foreach token generation step $t$ do
        foreach layer $l$ do
            foreach head $h$ do
                Compute full attention scores $\boldsymbol{\alpha}_{l,h}(\mathbf{q}_t, \mathbf{K}_{1:t})$
                foreach FC index $i$ do
                    Compute single-FC scores $\boldsymbol{\alpha}_{l,h}^{(i)}(\mathbf{q}_t, \mathbf{K}_{1:t})$
                    Calculate the CA score $\text{CA}_{\mathcal{K}}^{l,h,i}$ using Eq. 4
                    Store $\text{CA}_{\mathcal{K}}^{l,h,i}$ in $M[l][h][i]$
                end
            end
        end
    end
end
// Stage 2: Select Dominant FCs
Initialize an empty map $\overline{M}$ for mean CA scores
foreach $(l, h, i)$ in $M$ do
    $\overline{M}[l][h][i] \leftarrow \text{Mean}(M[l][h][i])$
end
$\mathcal{I}_{dom} \leftarrow$ TopK-Indices$(\overline{M}, k)$ // Select top-k indices based on $\overline{\text{CA}}$
return $\mathcal{I}_{dom}$
```

F LLM Usage

During the preparation of this manuscript, we utilized the AI-based language model ChatGPT, developed by OpenAI. Its use was strictly limited to language refinement, including grammar correction, stylistic enhancement, and rephrasing for clarity. All scientific concepts, experimental designs, data analyses, and conclusions presented herein are the original work of the authors and were conceived and executed without any substantive contribution from the language model.

Algorithm 2: Inference with FASA-M (Memory-Optimized Variant)

```
Input: Current query $\mathbf{q}_t$; Current key $\mathbf{k}_t$; Current value $\mathbf{v}_t$
Dominant FC indices $\mathcal{I}_{dom}$
Token budget $b$
Past KV cache: $C_{key}^{dom}$ (GPU), $C_{key}^{nondom}$ (CPU), $C_{val}$ (CPU)
Output: Next hidden state $\mathbf{h}_{t+1}$
Updated KV cache: $C_{key}^{dom}$, $C_{key}^{nondom}$, $C_{val}$
// Stage 1: Token Importance Prediction (TIP)
// Split key by dominant FCs
$\mathbf{k}_t^{dom}, \mathbf{k}_t^{nondom} \leftarrow \text{Split}(\mathbf{k}_t, \mathcal{I}_{dom})$
// Select corresponding query dimensions
$\mathbf{q}_t^{dom} \leftarrow \text{Select}(\mathbf{q}_t, \mathcal{I}_{dom})$
$K_{1:t}^{dom} \leftarrow \text{UpdateCache}(C_{key}^{dom}, \mathbf{k}_t^{dom})$
// Approximate scores using dominant parts
$\hat{\mathbf{S}}_t \leftarrow \mathbf{q}_t^{dom}(K_{1:t}^{dom})^\top$
// Identify indices of $b$ most salient tokens
$\mathcal{T}_t \leftarrow \text{TopK-Indices}(\hat{\mathbf{S}}_t, b)$
// Stage 2: Focused Attention Computation (FAC)
// Select dominant key parts on GPU
$K_{\mathcal{T}_t}^{dom} \leftarrow \text{SelectTokens}(K_{1:t}^{dom}, \mathcal{T}_t)$
// Update non-dominant cache on CPU
$C_{key}^{nondom} \leftarrow \text{UpdateCache}(C_{key}^{nondom}, \mathbf{k}_t^{nondom})$
$K_{1:t}^{nondom} \leftarrow \text{LoadFromCPU}(C_{key}^{nondom})$
// Select non-dominant key parts on CPU
$K_{\mathcal{T}_t}^{nondom} \leftarrow \text{SelectTokens}(K_{1:t}^{nondom}, \mathcal{T}_t)$
// Update value cache on CPU
$C_{val} \leftarrow \text{UpdateCache}(C_{val}, \mathbf{v}_t)$
$V_{1:t} \leftarrow \text{LoadFromCPU}(C_{val})$
// Select values on CPU
$V_{\mathcal{T}_t} \leftarrow \text{SelectTokens}(V_{1:t}, \mathcal{T}_t)$
// Offload required non-dominant keys to GPU
$K_{\mathcal{T}_t}^{nondom} \leftarrow \text{TransferToGPU}(K_{\mathcal{T}_t}^{nondom})$
// Offload required values to GPU
$V_{\mathcal{T}_t} \leftarrow \text{TransferToGPU}(V_{\mathcal{T}_t})$
// Reconstruct full keys for selected tokens
$K_{\mathcal{T}_t} \leftarrow \text{Combine}(K_{\mathcal{T}_t}^{dom}, K_{\mathcal{T}_t}^{nondom}, \mathcal{I}_{dom})$
// Compute full attention on the subset
$\boldsymbol{\alpha}_{\text{fac}} \leftarrow \text{Softmax}(\mathbf{q}_t K_{\mathcal{T}_t}^\top / \sqrt{d_k})$
$\mathbf{h}_{t+1} \leftarrow W_O(\boldsymbol{\alpha}_{\text{fac}} V_{\mathcal{T}_t})$
return $\mathbf{h}_{t+1}$ and updated caches
```