

Beyond Homogeneous Attention: Memory-Efficient LLMs via Fourier-Approximated KV Cache

Qiqi Wang^{1,2,3*}, Siyang He^{1,2,3*}, Xiaoran Liu^{1,2,3*}, Ruixiao Li^{1,2,3*},
Yuerong Song^{1,2,3}, Zhigeng Liu^{1,2,3}, Mianqiu Huang¹, Zengfeng Huang^{1,2},
Qipeng Guo², Ziwei He^{2,3†}, Xipeng Qiu^{1,2,3†}

¹Fudan University, ²Shanghai Innovation Institute, ³OpenMOSS Team

Correspondence: ziwei.he@sii.edu.cn, xpqiu@fudan.edu.cn

Abstract

Large Language Models struggle with memory demands from the growing Key-Value (KV) cache as context lengths increase. Existing compression methods homogenize head dimensions or rely on attention-guided token pruning, often sacrificing accuracy or introducing computational overhead. We propose **FourierAttention**, a training-free framework that exploits the heterogeneous roles of transformer head dimensions: lower dimensions prioritize local context, while upper ones capture long-range dependencies. By projecting the long-context-insensitive dimensions onto orthogonal Fourier bases, FourierAttention approximates their temporal evolution with fixed-length spectral coefficients. Evaluations on Llama and Qwen models show FourierAttention achieves the best long-context accuracy on LongBench and Needle-In-A-Haystack (NIAH). Besides, a custom Triton kernel, **FlashFourierAttention**, is designed to optimize memory via streamlined read-write operations, enabling efficient deployment without performance compromise.

1 Introduction

Large Language Models (LLMs) have achieved remarkable progress in text generation, comprehension, and reasoning (Sun et al., 2024; OpenAI, 2024; Guo et al., 2025). However, their autoregressive decoding relies on a Key-Value (KV) cache whose memory cost grows linearly with context length (Fu, 2024; Liu et al., 2025). This growing cache footprint becomes a major bottleneck for long-context inference, especially in resource-constrained deployment. Existing sparse attention and KV cache compression methods reduce memory usage, but often introduce accuracy degradation, attention recomputation, or additional system complexity (Cai et al., 2024; Yuan et al., 2025).

Existing training-free KV cache compression methods, like token eviction strategies (Xiao et al., 2024b; Zhang et al., 2023; Li et al., 2024b), prune sequence subsets but overlook the heterogeneous roles of head dimensions, leaving dimension-aware allocation largely unexplored. Similarly, hidden dimension compression (Chang et al., 2024; Saxena et al., 2024) methods apply uniform ratios, both neglect their distinct contribution across dimensions (Liu et al., 2024b; Peng et al., 2024). These approaches treat head dimensions as homogeneous, static units rather than dynamically allocating resources based on their importance.

Another limitation of existing methods lies in their reliance on attention-guided strategies (Zhang et al., 2023; Li et al., 2024b). Although these methods prune tokens with limited accuracy loss, they require attention scores for cache selection, introducing extra memory and latency overheads. Instead, we address this challenge by adapting the HiPPO perspective (Gu et al., 2020): temporal signals can be summarized as finite states under orthogonal bases (Gu et al., 2020; He et al., 2023). This motivates us to represent long-context-insensitive KV dimensions with fixed-length Fourier states, avoiding attention-guided cache selection and achieving both memory efficiency and computational efficiency.

Building on these insights, we introduce **FourierAttention**, a training-free KV cache compression framework based on the translated Fourier transform, shown in Figure 1. Unlike prior methods that process head dimensions uniformly, FourierAttention preserves long-context-sensitive dimensions and compresses others by a fixed set of Fourier coefficients ($N \ll L$). Compared with polynomial bases commonly used in HiPPO, Fourier bases offer shift-invariance and efficient temporal parallelism. During decoding, a customized Triton kernel **FlashFourierAttention** reconstructs KV chunks on the fly inside attention computation, re-

* Equal contribution.

† Corresponding Author.

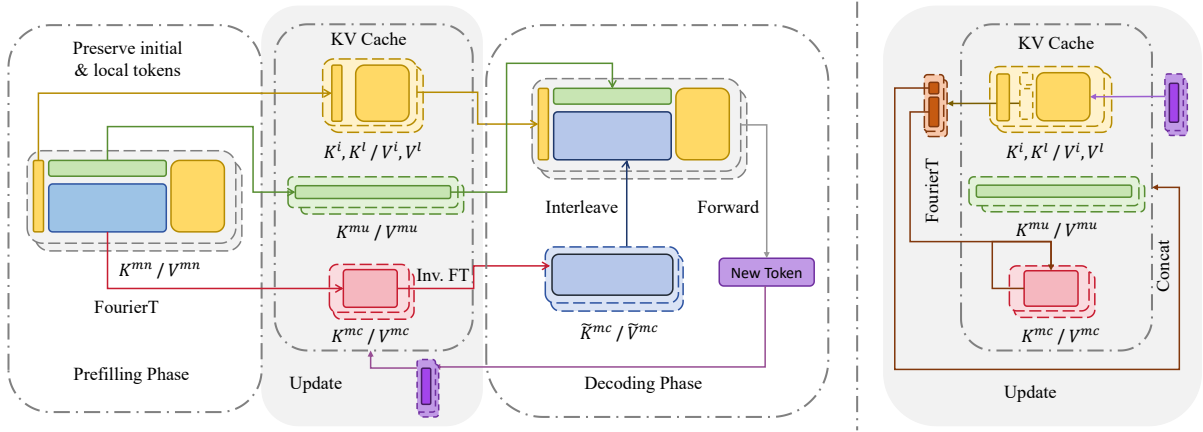


Figure 1: Overview of FourierAttention. During prefilling, we preserve the initial and local KV tokens, split the middle into retained and compressed dimensions, and project the compressed dimensions into fixed-length Fourier states. When new tokens arrive, tokens moving out of local window are incrementally folded into Fourier states.

ducing read-write overhead without materializing the full decompressed cache. Our contributions can be summarized as follows:

- We reveal head-dimension heterogeneity: lower dimensions capture local context, while upper dimensions model long-range dependencies, motivating our dimension-aware KV compression.
- We introduce FourierAttention, which optimizes KV cache by projecting its temporal evolution onto a fixed set of orthogonal Fourier bases. This method efficiently eliminates redundant components while preserving contextual fidelity, achieving a balance between memory and computational efficiency.
- Experiments on Llama and Qwen models over LongBench and NIAH show FourierAttention consistently outperforms strong KV-cache baselines with lower memory usage.

2 Related Work

KV cache optimization is a crucial technique for enhancing efficiency in attention-based LLMs (Fu, 2024; Liu et al., 2025). As context length increases, the KV cache in LLMs grows linearly, creating substantial memory overhead that becomes a bottleneck for long-context applications. Beyond architectural modifications during pretraining (Ainslie et al., 2023; Liu et al., 2024a), existing training-free optimization approaches mainly involve token eviction or compression. The former discards tokens based on positional or attention patterns, including StreamingLLM (Xiao et al., 2024b), H2O (Zhang

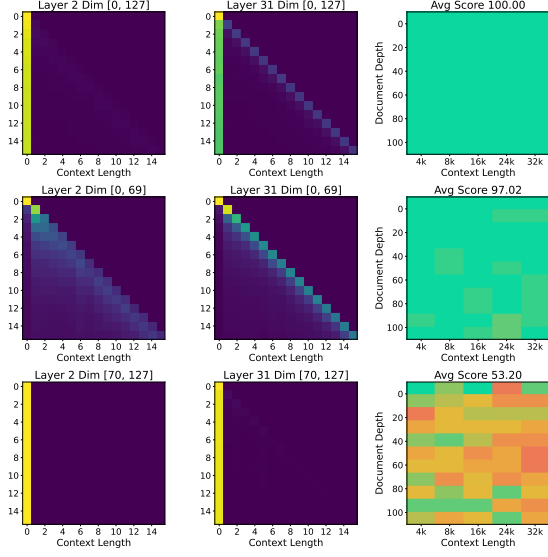
et al., 2023), SnapKV (Li et al., 2024b), PyramidKV (Cai et al., 2024), FastKV (Jo et al., 2025) and LaCache (Shi et al., 2025), while the latter compresses KV cache through quantization or low-rank projection, such as KIVI (Liu et al., 2024c), KVQuant (Hooper et al., 2024), and Palu (Chang et al., 2024). However, these methods lack fine-grained consideration of different head dimensions in the KV cache, applying uniform optimization across all dimensions. In contrast, our FourierAttention compresses most dimensions to a fixed length while preserving long-context-sensitive dimensions, effectively reducing KV cache size while maintaining the original long-context capabilities.

3 Methodology

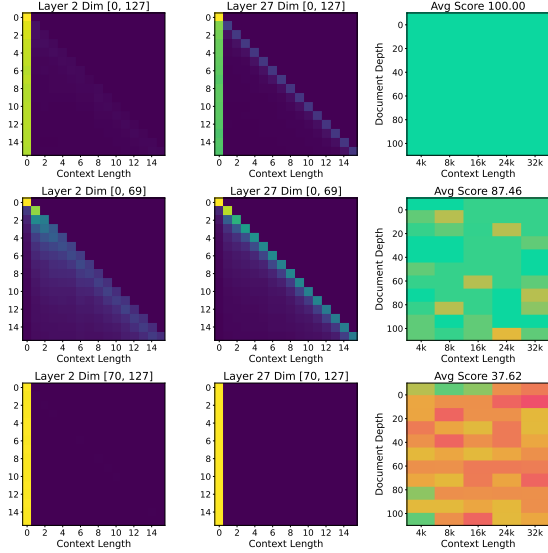
3.1 Head Dimension Heterogeneity

We analyze the heterogeneous sensitivity of transformer head dimensions to varying context lengths. By visualizing attention scores across 128 dimensions in Llama architecture (Figure 2), we identify a bifurcation in attention patterns: 0-69 dimensions exhibit sharp focus on short-range context, with score distributions concentrated on recent tokens, while 70-127 dimensions maintain a persistent bias toward initial “sink tokens”.

This contiguous dimension split is not accidental but closely relates to the characteristics of Rotary Positional Embeddings (RoPE). Specifically in Llama architecture, index 70 corresponds to the “critical dimension” d_{extra} (Liu et al., 2024b), which is calculated based on the head dimension d , the initial pre-training length T_{train} , and the initial rotary



(a) Llama 3.1-8B.



(b) Llama 3.2-3B.

Figure 2: Visualization of average attention-score components and NIAH perturbation results in Llama 3.1-8B and Llama 3.2-3B over 32 sentences with 16 tokens. Lower dimensions (0–69) align with the local branch, while upper dimensions (70–127) correspond to the global branch. Adding Gaussian noise to lower dimensions has little effect, whereas perturbing upper dimensions significantly degrades NIAH performance.

base β as below and detailed in Appendix A.

$$d_{\text{extra}} = 2 \left\lceil \frac{d}{2} \log_{\beta} \frac{T_{\text{train}}}{2\pi} \right\rceil \quad (1)$$

Head dimensions below d_{extra} are trained with complete positional information and are locally monotonic, biasing them toward high-frequency

local features. In contrast, dimensions above d_{extra} have incomplete positional information and longer monotonic intervals, focusing on low-frequency global features (Wei et al., 2025; Barbero et al., 2024). This divergence suggests distinct contextual roles in head dimensions, where specialized subsets prioritize local versus global signal retention.

To further validate the dimension bifurcation hypothesis, we conduct a Needle-In-A-Haystack retrieval test with sequences up to 32K tokens. As shown in Figure 2, both unperturbed models achieve an average score of 100.00. For Llama 3.1-8B, adding Gaussian noise to the first 70 dimensions only slightly reduces the score to 97.02, whereas perturbing the latter 58 dimensions causes a substantial drop to 53.20. An even more pronounced trend is observed on Llama 3.2-3B: perturbing the first 70 dimensions yields a score of 87.46, while perturbing the latter 58 dimensions reduces it to 37.62, with failures across different retrieval depths and context lengths. These results align with the critical-dimension analysis above, indicating that upper dimensions are more important for long-range information retention, while lower dimensions mainly encode local context. Motivated by this heterogeneity, FourierAttention compresses long-context-insensitive KV dimensions into fixed-length states via the translated Fourier transform, with compression-friendly dimensions selected by reconstruction fidelity in Section 3.4.

3.2 Preliminary: HiPPO Framework

Inspired by HiPPO (Gu et al., 2020), we compress these less context-sensitive dimensions into fixed-length states to reduce KV cache storage. Under the HiPPO framework, an infinitely long sequence, $f_{1 \dots L}$, can be approximated by finite-length states, $c \in \mathbb{R}^N$, as the combining coefficients of finite-order basis functions. HiPPO designs different state update equations for various basis functions under different measure functions, such as LegT based on Legendre Polynomials in a translated fixed window size. Among these methods, FourierT based on Translated Fourier Transform is most suitable for token-wise parallelism in transformers, because it can be expressed in matrix form and performed independently in different order states. Therefore, we adopt FourierT to compress cache, $K, V \in \mathbb{R}^{L \times d}$, which also achieves better downstream performance in Section 5.1.

3.3 Online Compression via HiPPO-FourierT

Frequency-domain operators have been increasingly explored in large language model inference (He et al., 2023, 2026). Here, we use HiPPO-FourierT for online KV-cache compression. We set the translated window length in FourierT to maximum context length, ensuring effective compression within valid input-output ranges. In the prefilling phase, we preserve all dimensions of the initial L_{init} and the local L_{local} tokens,

$$\begin{aligned} \mathbf{K}^i, \mathbf{K}^l &= \mathbf{K}[:, L_{\text{init}}], \mathbf{K}[-L_{\text{local}} :] \\ \mathbf{V}^i, \mathbf{V}^l &= \mathbf{V}[:, L_{\text{init}}], \mathbf{V}[-L_{\text{local}} :] \end{aligned} \quad (2)$$

and distinguish the dimension indices $\mathcal{D}^{ku}$, $\mathcal{D}^{kc}$, $\mathcal{D}^{vu}$, $\mathcal{D}^{vc}$ in KV cache for uncompressing and compressing, to enable training-free integration.

$$\begin{aligned} \mathbf{K}^{mn} &= \mathbf{K}[L_{\text{init}} : -L_{\text{local}}, \mathcal{D}^{kc}], \\ \mathbf{V}^{mn} &= \mathbf{V}[L_{\text{init}} : -L_{\text{local}}, \mathcal{D}^{vc}], \\ \mathbf{K}^{mu} &= \mathbf{K}[L_{\text{init}} : -L_{\text{local}}, \mathcal{D}^{ku}], \\ \mathbf{V}^{mu} &= \mathbf{V}[L_{\text{init}} : -L_{\text{local}}, \mathcal{D}^{vu}]. \end{aligned} \quad (3)$$

We preserve $\mathbf{K}^{mu}$, $\mathbf{V}^{mu}$, compress $\mathbf{K}^{mn}$, $\mathbf{V}^{mn}$ to fixed-size $\mathbf{K}^{mc} \in \mathbb{R}^{2N \times |\mathcal{D}^{kc}|}$, $\mathbf{V}^{mc} \in \mathbb{R}^{2N \times |\mathcal{D}^{vc}|}$ and use the original KV for forward propagation.

$$\begin{aligned} \mathbf{K}^{mc} &= \frac{1}{T} \mathcal{F} \mathbf{K}^{mn}, \quad \mathbf{V}^{mc} = \frac{1}{T} \mathcal{F} \mathbf{V}^{mn}, \\ \mathbf{O} &= \text{flash_attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}). \end{aligned} \quad (4)$$

We detail the mathematical derivation in Appendix B. For the compression matrix in FourierT, originally, $\mathcal{F} \in \mathbb{C}^{N \times L_{\text{middle}}}$, where $L_{\text{middle}} = L - L_{\text{init}} - L_{\text{local}}$ and $\mathcal{F}_{nt} = e^{i \frac{2\pi n t}{T}}$. However, since caches in mainstream LLMs are real-valued, we convert complex numbers to 2D vectors, transforming N -order complex states into $2N$ -order real states. We denote $N' = 2N$, and the real compression matrix in FourierT is $\mathcal{F} \in \mathbb{R}^{N' \times L_{\text{middle}}}$ as shown in Equation 5.

$$\mathcal{F} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ 1 & \cos \frac{2\pi}{T} & \cdots & \cos \frac{2\pi(L_{\text{middle}}-1)}{T} \\ 0 & \sin \frac{2\pi}{T} & \cdots & \sin \frac{2\pi(L_{\text{middle}}-1)}{T} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \cos \frac{2\pi(N-1)}{T} & \cdots & \cos \frac{2\pi(N-1)(L_{\text{middle}}-1)}{T} \\ 0 & \sin \frac{2\pi(N-1)}{T} & \cdots & \sin \frac{2\pi(N-1)(L_{\text{middle}}-1)}{T} \end{bmatrix} \quad (5)$$

In the decoding phase, FourierAttention reconstructs intermediate cache $\tilde{\mathbf{K}}^m, \tilde{\mathbf{V}}^m$ via inverse

Fourier transform in attention computation with the current query vector $\mathbf{q}_{t+1}$,

$$\begin{aligned} \tilde{\mathbf{K}}^m[\mathcal{D}^{ku}] &\leftarrow \mathbf{K}^{mu}, \quad \tilde{\mathbf{K}}^m[\mathcal{D}^{kc}] \leftarrow \mathcal{F}^T \mathbf{K}^{mc} \\ \tilde{\mathbf{V}}^m[\mathcal{D}^{vu}] &\leftarrow \mathbf{V}^{mu}, \quad \tilde{\mathbf{V}}^m[\mathcal{D}^{vc}] \leftarrow \mathcal{F}^T \mathbf{V}^{mc} \\ \tilde{\mathbf{K}} &= \text{cat}(\mathbf{K}^i, \tilde{\mathbf{K}}^m, \mathbf{K}^l), \quad \tilde{\mathbf{V}} = \text{cat}(\mathbf{V}^i, \tilde{\mathbf{V}}^m, \mathbf{V}^l) \\ \mathbf{o}_{t+1} &= \text{flash_attention}(\mathbf{q}_{t+1}, \tilde{\mathbf{K}}, \tilde{\mathbf{V}}). \end{aligned} \quad (6)$$

and compresses tokens leaving the local window.

$$\begin{aligned} \mathbf{K}^{mc} &\leftarrow \mathbf{K}^{mc} + \frac{1}{T} \mathbf{f}_{L_{\text{middle}}+1} \mathbf{K}^l[0, \mathcal{D}^{kc}] \\ \mathbf{V}^{mc} &\leftarrow \mathbf{V}^{mc} + \frac{1}{T} \mathbf{f}_{L_{\text{middle}}+1} \mathbf{V}^l[0, \mathcal{D}^{vc}] \\ \mathbf{f}_{t+1} &= \begin{bmatrix} 1 & 0 & \cdots & \cos \frac{2\pi(N-1)L_{\text{middle}}}{T} & \sin \frac{2\pi(N-1)L_{\text{middle}}}{T} \end{bmatrix}^T \end{aligned} \quad (7)$$

For notational clarity, Eq. 6 presents full-cache decompression. In practice, our Triton kernel FlashFourierAttention fuses inverse Fourier reconstruction with FlashDecoding, reconstructing only block-wise K/V chunks in SRAM for attention. Thus, the full decompressed KV cache is never materialized in HBM, preserving the memory benefit. Details are provided in Appendix G.

3.4 Fine-Grained Compression Schema

To select compression-friendly KV dimensions, we perform a lightweight offline calibration on the 32K NeedleBench dataset (Li et al., 2024a). We compress and reconstruct KV states extracted from forward passes, and use mean-squared error (MSE) to measure reconstruction fidelity. Lower MSE values identify inherently "smoother" dimensions, indicating their suitability for our FourierAttention.

Based on further KV-cache observations, we adopt a fine-grained asymmetric inverted-pyramid compression schema. We analyze the temporal standard deviation of each KV dimension across layers, and find that K states generally vary more than V states, while upper layers vary more than lower layers in both Llama 3.1-8B and Llama 3.2-3B (Figure 3). Accordingly, we compress more dimensions in the smoother V cache and lower layers, while retaining more K dimensions and upper-layer dimensions in their original temporal form.

Interestingly, this differs from most KV cache compression approaches. Works like Cai et al. (2024) and Xing et al. (2024) suggest preserving more KV caches in lower layers, as attention becomes sparser in upper layers. However, in FourierAttention, the optimization criterion is whether the

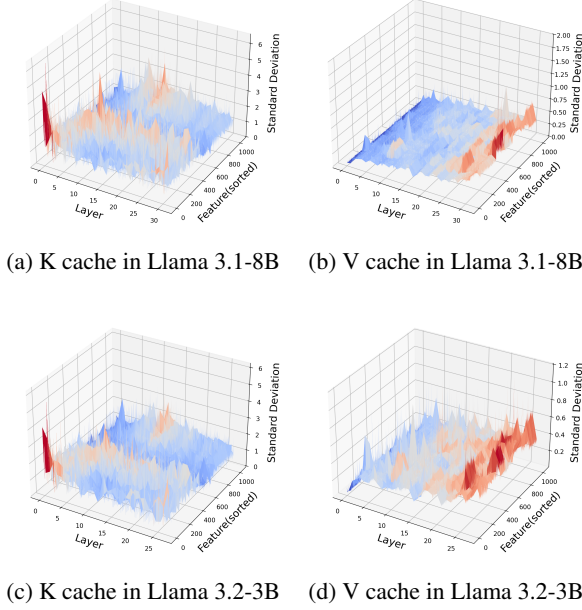


Figure 3: Temporal standard deviation of the K and V caches across layers in Llama 3.1-8B and Llama 3.2-3B. Feature dimensions are arranged according to their original indices within each attention head. In both models, K states generally exhibit greater temporal variation than V states, and the variation tends to increase in upper layers.

dimension can be well reconstructed. Since caches in upper layers exhibit more oscillatory features due to more deterministic predictions, we retain more dimensions to maintain output stability.

4 Experiment

4.1 Setup

We conduct experiments on Llama 3.2-3B, Llama 3.2-1B (Meta, 2024a), Llama 3.1-8B (Dubey et al., 2024), Qwen2.5-1.5B, and Qwen2.5-3B (Qwen et al., 2024). By default, FourierAttention uses $L_{\text{init}} = 4$, $L_{\text{local}} = 1024$, and $N = 512$ Fourier states, corresponding to $N' = 1024$ real-valued states. The translated Fourier window is set to cover the effective input-output range, i.e., $T_{\text{window}} = \text{prompt_length} + \text{max_out_len}$.

For dimension selection, we perform a lightweight offline calibration on the prompt portion of the 32K Needle-In-A-Haystack benchmark in OpenCompass (Contributors, 2023), and select dimensions with lower KV reconstruction error as compression-friendly dimensions. Following the asymmetric inverted-pyramid schema introduced in Section 3.4, we compress 90%/95% of K/V dimensions in the first 14% layers, 50%/70% in the

last 28% layers, and 80%/80% in the remaining layers. Overall, 76% of KV cache dimensions are compressed into fixed-length Fourier states. We further analyze the robustness of this percentage-based schema in Section 4.4, showing that moderate perturbations to the compression ratios remain acceptable. All experiments are performed on an NVIDIA H100 GPU with FP16 precision and accelerated with FlashAttention2 (Dao, 2024).

4.2 Long-Context Evaluation

We evaluate FourierAttention on LongBench (Bai et al., 2023) and Needle-In-A-Haystack (NIAH) (Kamradt, 2023; Li et al., 2024a) in OpenCompass (Contributors, 2023), with a truncation context length of 32K. We compare with StreamingLLM (Xiao et al., 2024b), SnapKV (Li et al., 2024b), PyramidKV (Cai et al., 2024), FastKV (Jo et al., 2025), LaCache (Shi et al., 2025), and Palu (Chang et al., 2024), covering token eviction and feature compression.

We align baselines to the same effective KV-cache budget using each method’s native budget control. Since FourierAttention compresses 76% of KV dimensions to fixed Fourier states, fully reserving initial and local tokens, it yields an effective compression ratio of 71.1% at 32K context length (Appendix C.1). Accordingly, SnapKV and PyramidKV retain 28.9% of KV positions, while StreamingLLM keeps 4 initial tokens and adjusts its local window to the same budget. We set LaCache’s layer-span budget to 0.29, FastKV’s KV retention rate to 30%, and Palu’s feature-compression ratio to 70%, the closest available settings.

For LongBench as shown in Tables 1 and 6, FourierAttention consistently achieves the best average performance among cache-optimized methods across Llama and Qwen models, remaining closest to full attention. On NIAH (Figures 4 and 9), FourierAttention also preserves retrieval accuracy better than baselines at 32K context. While SnapKV and PyramidKV still exhibit recall errors and Palu degrades under high compression, FourierAttention maintains long-context capability by preserving long-context-sensitive dimensions.

We further evaluate different values of FourierAttention’s N' on Llama 3.2-3B, with SnapKV configured under the same corresponding effective KV budget. As shown in Figure 5, FourierAttention robustly outperforms SnapKV across different budgets, and we use $N' = 1024$ as the default setting to balance performance and speed.

Table 1: Results of Llama Series (Dubey et al., 2024; Meta, 2024b) and Qwen Series (Qwen et al., 2024) on LongBench (Bai et al., 2023). Our FourierAttention (FA) achieves the best average performance among cache-optimized methods, including StreamingLLM (SLM), SnapKV, PyramidKV, FastKV, LaCache, and Palu, and shows the closest performance to LLMs with full attention.

	Single-Doc			Multi-Doc			Summary			Few-shot			Synthetic		Code		Avg.
	NQ	Qsp	MF	HQ	WQ	Msq	GR	QS	MN	TR	TQ	SS	PC	PR	LCC	Re-P	
Llama-3.2-3B	10.3	21.7	35.5	9.6	12.8	6.8	30.2	23.8	28.2	70.0	87.2	38.2	0.0	7.0	70.0	66.4	38.0
+ SLM	9.1	17.6	21.6	7.1	9.8	4.0	19.0	21.3	23.2	53.0	84.3	39.8	1.4	6.5	55.5	53.5	31.2
+ SnapKV	9.5	21.6	35.5	9.5	12.8	6.6	30.3	23.5	28.2	70.0	86.3	38.4	0.0	6.3	58.8	57.1	<u>35.2</u>
+ PyramidKV	9.5	21.6	35.5	9.5	12.8	6.6	30.3	23.5	28.2	70.0	86.4	38.3	0.0	6.4	58.7	57.1	35.1
+ FastKV	9.8	20.1	33.6	9.5	14.8	6.7	15.8	13.7	9.7	71.0	87.6	30.3	0.0	7.0	68.8	65.2	35.0
+ LaCache	9.7	21.4	35.6	9.6	12.9	6.8	19.1	14.4	15.9	70.0	87.2	29.9	0.0	7.0	58.8	57.4	33.1
+ Palu	2.0	19.2	20.4	5.8	10.3	2.7	13.4	4.1	14.0	57.0	47.4	21.5	1.5	3.1	55.1	49.5	25.5
+ FA (ours)	12.8	21.1	35.8	9.8	11.6	6.5	23.0	23.8	24.1	69.0	87.0	39.4	0.0	6.2	69.8	63.3	37.0
Llama-3.1-8B	13.2	20.2	32.8	12.0	13.6	8.7	29.7	25.1	0.9	73.5	91.0	47.3	0.8	26.8	72.0	69.3	39.5
+ SLM	7.9	13.9	15.6	7.8	10.1	4.5	19.9	21.5	9.9	61.5	84.7	43.5	1.3	6.2	58.4	56.4	31.5
+ SnapKV	20.5	20.2	32.7	12.0	13.6	8.6	30.3	24.9	12.8	73.5	90.0	46.8	0.8	26.4	60.4	59.9	37.6
+ PyramidKV	20.5	20.2	32.7	12.0	13.6	8.6	30.3	24.9	12.8	73.5	90.0	46.8	0.8	26.4	60.4	59.8	37.6
+ FastKV	19.3	19.3	31.4	12.1	14.2	8.2	16.5	12.6	0.4	74.5	92.0	38.2	0.9	30.9	70.7	68.1	<u>37.9</u>
+ LaCache	21.4	20.2	32.4	12.0	13.8	8.7	18.5	13.1	0.6	73.5	91.0	37.8	0.8	26.8	60.5	60.2	35.4
+ Palu	4.5	18.0	21.6	9.5	11.3	5.2	17.3	6.9	9.0	68.5	83.4	32.3	0.6	14.6	56.7	54.6	30.7
+ FA (ours)	15.6	19.8	32.7	12.0	13.5	7.9	24.1	24.0	0.7	73.0	91.2	46.0	1.1	26.9	71.4	66.3	38.7
Qwen2.5-1.5B	10.6	24.5	36.6	23.4	21.8	11.0	30.2	23.5	26.1	71.5	77.9	42.9	2.0	7.5	56.1	58.8	36.7
+ SLM	12.7	13.6	26.1	10.2	14.6	5.3	21.5	20.4	23.9	56.5	56.6	40.5	1.4	5.8	48.9	46.9	28.9
+ SnapKV	8.8	24.7	36.6	22.5	21.8	10.5	30.2	23.1	26.1	71.5	77.2	42.6	2.0	7.5	45.6	50.3	33.9
+ PyramidKV	8.8	24.7	36.6	22.3	21.8	10.5	30.2	23.1	26.1	71.5	77.2	42.5	2.0	7.5	45.7	50.4	<u>33.9</u>
+ FastKV	12.5	21.4	30.5	24.3	20.3	9.6	14.4	17.5	13.1	62.0	76.8	31.3	2.0	6.0	53.5	56.0	32.4
+ LaCache	11.6	24.0	32.8	25.6	22.0	10.9	17.4	17.2	15.7	71.5	77.5	31.7	2.0	9.5	45.8	50.4	32.1
+ Palu	9.1	19.6	37.5	10.5	19.8	4.3	16.8	12.6	12.4	70.0	68.3	22.2	1.6	3.5	47.2	46.5	28.4
+ FA (ours)	6.7	24.2	34.4	22.9	21.8	10.3	21.6	23.0	25.0	63.5	76.5	42.0	2.0	8.5	55.9	55.4	34.8
Qwen2.5-3B	14.3	29.9	45.9	38.4	33.5	20.6	29.7	24.2	25.2	69.0	89.5	45.3	2.0	8.0	69.3	65.6	42.7
+ SLM	13.9	17.6	25.6	21.3	23.6	7.8	22.3	19.9	24.5	51.5	79.1	42.4	1.5	6.5	56.0	52.7	33.2
+ SnapKV	14.5	29.9	45.9	37.5	34.0	19.8	29.4	23.8	25.2	69.0	88.6	44.9	1.5	8.0	57.8	55.6	<u>39.7</u>
+ PyramidKV	14.5	29.9	45.9	37.4	34.0	19.8	29.3	23.8	25.2	69.0	89.0	44.8	1.5	8.0	57.8	55.6	<u>39.7</u>
+ FastKV	16.8	19.1	36.1	39.2	30.3	19.8	13.8	17.9	13.6	59.5	87.9	33.5	2.0	8.0	65.0	63.8	37.9
+ LaCache	18.0	29.9	42.8	37.1	33.0	20.6	17.5	17.8	15.9	69.0	89.4	34.6	1.5	8.5	57.8	56.4	37.9
+ Palu	6.3	15.5	35.2	11.3	17.3	5.5	17.6	11.4	14.2	68.0	78.3	26.1	4.7	5.0	58.1	55.3	31.5
+ FA (ours)	6.4	27.7	43.3	30.4	31.7	16.3	22.0	23.4	24.7	66.5	88.9	45.2	2.0	7.5	68.9	62.0	40.1

4.3 Efficiency Validation

In addition to downstream performance, we compare FourierAttention with SnapKV and Palu in terms of memory usage and prefill latency, as they represent token eviction and feature compression, respectively. By fusing Fourier reconstruction with FlashDecoding, our custom Triton kernel, Flash-FourierAttention (Appendix G), avoids materializing the reconstructed KV cache in HBM and substantially reduces intermediate storage and memory traffic. As shown in Figure 6, this advantage becomes increasingly pronounced with longer contexts, making FourierAttention the only method that supports 80K inference on both Llama 3.1-8B and Llama 3.2-3B using a single H100. Since SnapKV requires full attention during prefill, it may encounter OOM errors in long-context settings.

Table 2: Prefill latency (s) for SnapKV, Palu, and FourierAttention on the Llama series. OOM denotes out-of-memory.

Method	8K	16K	32K	48K	64K	80K
Llama 3.1-8B Prefill Latency (s)						
SnapKV	0.29	0.66	1.65	3.06	OOM	OOM
Palu	0.31	0.69	1.81	3.33	5.23	OOM
FA (ours)	0.37	0.76	1.80	3.30	5.04	7.29
Llama 3.2-3B Prefill Latency (s)						
SnapKV	0.15	0.36	0.93	1.75	OOM	OOM
Palu	0.16	0.38	0.99	1.88	3.00	4.32
FA (ours)	0.21	0.44	1.03	1.95	2.99	4.44

Prefill latency across methods are shown in Table 2. Our FFT-based Fourier transform remains competitive with SnapKV and Palu whenever they fit in memory, and matches or slightly outperforms Palu in several long-context settings.

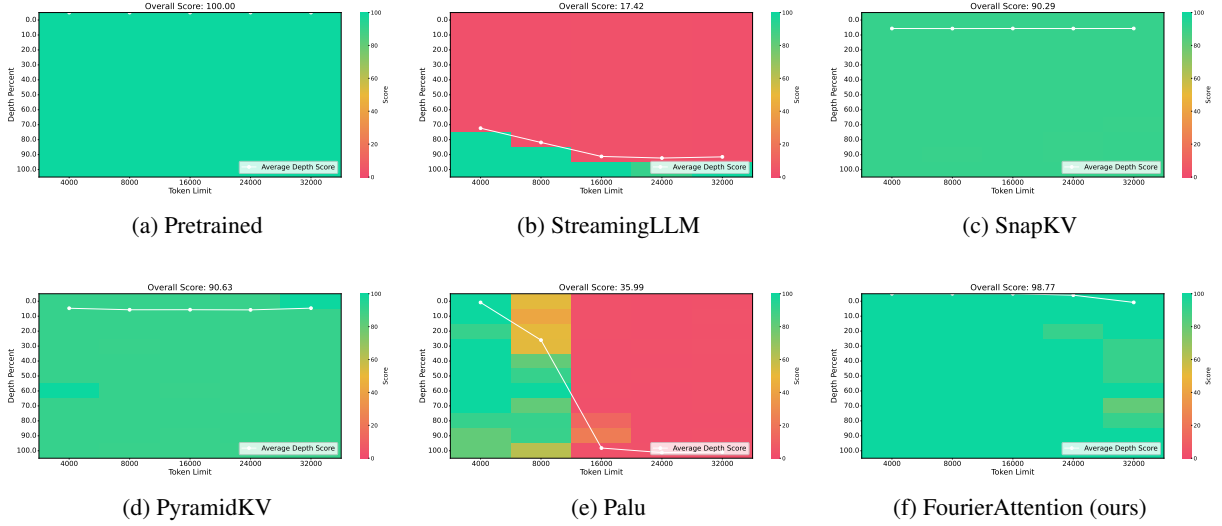


Figure 4: Results of Llama 3.2-3B on Needle-In-A-Haystack (Kamradt, 2023). FourierAttention achieves the highest average score over all baselines, and shows the closest performance with LLMs with full attention.

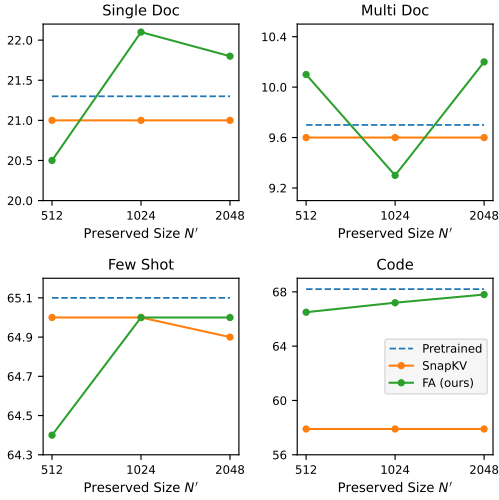
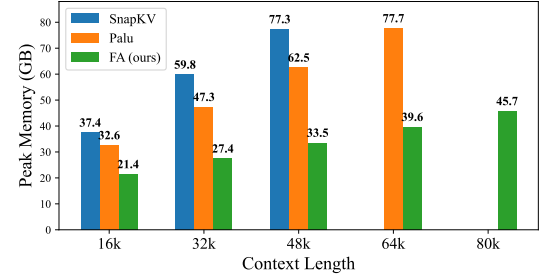


Figure 5: Comparison between FourierAttention and SnapKV on Llama 3.2-3B under matched effective KV budgets for each N' setting of FourierAttention.

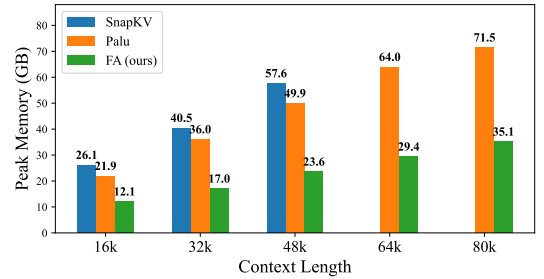
4.4 Stability of Grouping Configuration

We evaluate the robustness of FourierAttention’s configuration on Llama 3.2-3B and Qwen2.5-3B (detailed in Appendix C). Variants include shifting the boundary between low and high layers by $\pm 7\%$ of total layers (dis1/dis2), inversely adjusting the compression ratios for K and V in low/high layers by $\pm 5\%$ (dis3/dis4) or slightly perturbing them for dis5, and applying uniform compression across all layers and between K/V (uniform2), while keeping the total memory budget fixed.

Table 8 and Table 9 in Appendix C summarizes the results for all LongBench subtasks. All per-



(a) Memory cost of Llama 3.1-8B



(b) Memory cost of Llama 3.2-3B

Figure 6: Memory comparison across context lengths from 8K to 80K.

turbed variants remain above PyramidKV in most tasks and average score, even the naive uniform schema, confirming the robustness of our design.

5 Discussion

5.1 Choice of Basis Functions

Although FourierAttention employs HiPPO-FourierT for compression, polynomial bases such as LegT have been proposed in Gu et al. (2020) with potentially superior reconstruction. We compare LegT and FourierT on the same selected

Table 3: Validation of basis function and compression schema in Llama 3.2-3B based on RULER in 4K context length. Besides AGG tasks, FourierAttention achieves performance close to original model.

	NIAH								AGG			QA		Avg.
	SK1	SK2	SK3	MK1	MK2	MK3	MV	MQ	CWE	FWE	VT	SQ	HP	
Llama 3.2-3B	100.0	100.0	100.0	99.0	100.0	99.0	100.0	99.8	60.0	89.7	97.0	77.0	53.0	90.3
+ FourierT	100.0	100.0	98.0	99.0	99.0	100.0	96.0	97.8	57.9	80.7	76.6	81.0	51.0	87.5
+ LegT	55.0	82.0	42.0	89.0	93.0	50.0	93.8	84.0	13.5	58.3	76.0	36.0	21.0	61.0
+ uniform	100.0	100.0	99.0	99.0	99.0	98.0	93.0	98.8	59.7	77.3	78.2	80.0	50.0	87.1
+ KV inv.	100.0	100.0	100.0	99.0	99.0	100.0	93.5	98.8	54.3	79.7	72.6	80.0	53.0	86.9
+ layer inv.	100.0	98.0	88.0	98.0	97.0	94.0	80.0	96.8	41.3	69.3	61.4	80.0	50.0	81.1
+ random10%	90.0	87.0	35.0	82.0	69.0	50.0	52.8	63.3	5.2	55.0	28.8	81.0	50.0	57.6

dimensions with identical compression schema. Figure 10 illustrates reconstruction loss on 4 randomly selected KV dimensions from layer 0 of Llama 3.2; FourierT consistently achieves lower reconstruction error(MSE).

We further evaluate FourierT and LegT compression on Llama 3.2-3B using more discriminative RULER benchmark (Hsieh et al., 2024) in 4K context. Results in Table 3 show FourierT still performs better, demonstrating that FourierT offers better parallelizability for compression efficiency and performance in downstream evaluation.

5.2 Validation of the Asymmetric Inverted-Pyramid Schema

As mentioned in Section 3.4, we propose a more fine-grained compression scheme based on additional observations of the KV cache. As shown in Table 3, we compare four approaches: uniform compression across all layers and between KV (uniform), inverted KV compression schema by K-priority over V (KV inv.), inverted layer-wise compression schema by upper-layer priority over lower-layer (layer inv.) and randomly swap 10% of the identified non-sensitive dimensions with sensitive ones within each layer. Results demonstrate that our original V-priority and lower-layer-priority compression schema achieves superior performance on RULER 4K. Meanwhile, the perturbation with 10% error rate leads to severe degradation, which strongly demonstrate the high importance of our dimension selection scheme.

5.3 Compressed Dimension Distribution

We analyze the dimension-selection frequencies of FourierAttention, averaged across heads and grouped into 16-dimension bins. Results in Figure 7 show that in both Llama 3.1-8B and Llama 3.2-3B, starting from layer 2, lower dimensions are more frequently compressed while the upper di-

mensions tend to preserve complete temporal information in our FourierAttention. This phenomenon is more evident in upper layers, where fewer dimensions are chosen to be compressed. As illustrated in Figure 2, these uncompressed upper dimensions primarily contribute to forming attention sinks and capturing long-context semantic relationships, thus requiring complete retention, whereas other dimensions can be stored with limited length.

5.4 Comparison with Head-Level Methods

Although FourierAttention operates at the dimension level, long-context-sensitive dimensions are unevenly distributed across attention heads, suggesting a connection to head-level retrieval mechanisms. We compare the heads induced by our sensitive-dimension counts with the retrieval heads identified by DuoAttention (Xiao et al., 2024a) and HeadKV (Fu et al., 2024). Across three model families, the Jaccard overlaps reach 0.713–0.884 and 0.725–0.875, respectively, indicating that retrieval-head importance may arise from concentrated sensitive dimensions. Detailed results are reported in Table 11 Appendix E.

To isolate the effect of compression granularity, we construct a head-level variant named HeadFA. In each layer, HeadFA fully preserves the heads containing the largest numbers of sensitive dimensions and applies Fourier compression to all dimensions in the remaining heads. The number of preserved heads is chosen to match the uncompressed-dimension ratio of FourierAttention, ensuring a comparable cache budget.

As shown in Table 4, HeadFA consistently underperforms FourierAttention across all task categories under the same KV-cache budget. This result indicates that preserving entire heads cannot fully capture the fine-grained heterogeneity within each head, supporting dimension-level allocation as a more effective compression granularity.

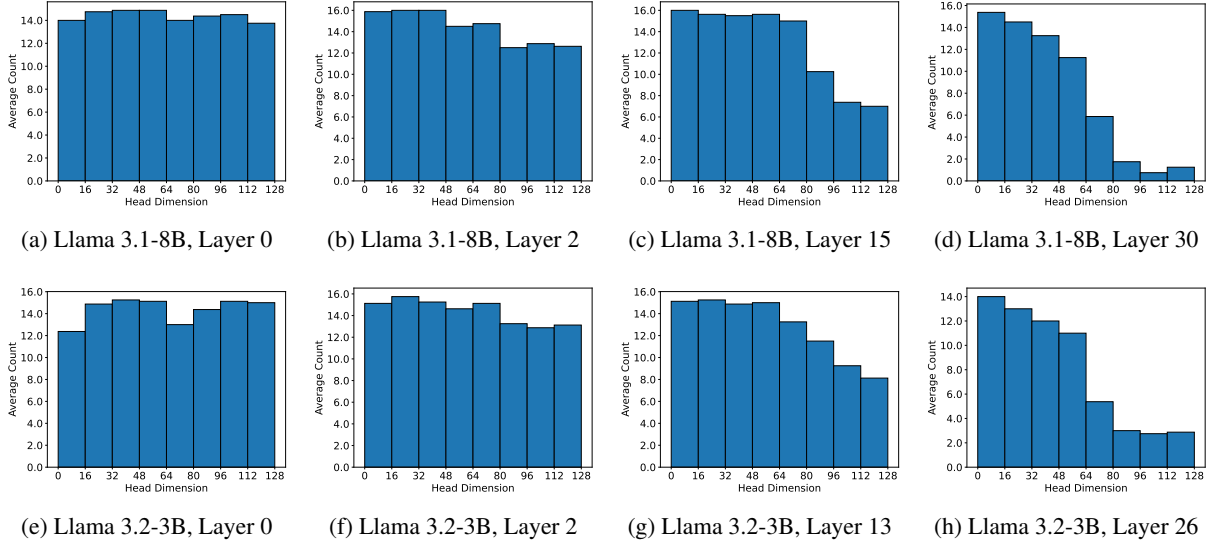


Figure 7: Distribution of dimensions selected for compression in representative layers of Llama 3.1-8B and Llama 3.2-3B. The statistics are averaged across attention heads and grouped every 16 dimensions.

Table 4: LongBench comparison between dimension-level FourierAttention and head-level HeadFA under matched compression budgets on Llama 3.2-3B.

Method	SD	MD	Sum	FS	Syn.	Code	Avg.
Llama 3.2-3B	21.3	9.7	27.4	65.1	3.5	68.2	38.0
+ HeadFA	18.7	9.1	22.7	64.2	2.5	65.2	35.7
+ FA (ours)	22.1	9.3	23.6	65.1	3.1	66.6	37.0

5.5 Integration with Low-Bit Quantization

Beyond its standalone use, FourierAttention can potentially be combined with orthogonal KV-cache compression techniques. While FourierAttention reduces temporal redundancy by representing selected dimensions with fixed-length Fourier states, low-bit quantization reduces the numerical precision of stored elements, making it a natural complement. We therefore integrate FourierAttention with KIVI 4-bit quantization and evaluate the combination on Llama 3.2-3B under identical settings. As shown in Table 12, FA+KIVI maintains performance close to both standalone methods across LongBench task categories. Table 5 further shows that FA+KIVI achieves the lowest persistent KV-cache footprint and reserved GPU memory at every tested context length from 8K to 80K, with the advantage becoming more pronounced at longer contexts. Detailed integration procedures are provided in Appendix F. Overall, these results demonstrate Fourier compression and low-bit quantization can be effectively combined to further reduce memory usage without substantial performance degradation.

Table 5: Persistent KV-cache footprint and reserved GPU memory (GB) of FourierAttention (FA), KIVI, and FA+KIVI on Llama 3.2-3B. Lower is better.

Method	8K	16K	32K	48K	64K	80K
Persistent KV-Cache Footprint (GB)						
FA	0.448	0.645	1.038	1.420	1.824	2.217
KIVI	0.261	0.517	1.030	1.543	2.055	2.568
FA+KIVI	0.256	0.317	0.440	0.563	0.686	0.809
Reserved GPU Memory (GB)						
FA	9.20	12.08	17.03	23.57	29.36	35.15
KIVI	11.80	13.92	21.13	31.49	44.76	41.25
FA+KIVI	7.79	10.53	12.57	16.87	20.66	24.95

6 Conclusion

We propose FourierAttention, a KV cache optimization approach that compresses long-context-insensitive dimensions without sacrificing contextual awareness based on a phenomenon in transformer head dimensions: lower dimensions capture local features, while upper ones capture long-context dependencies. Inspired by HiPPO, we optimize the long-context-insensitive KV cache through a translated Fourier transform into fixed-length states in prefilling phase and reconstruct the KV cache in decoding phase. FourierAttention shows the best performance on the Llama and Qwen Series in LongBench and NIAH on average and outperforms head-level compression. We also improve the efficiency of FourierAttention through a customized Triton-based kernel, FlashFourierAttention, eliminating intermediate read-write operations and reducing memory overhead.

Limitations

Although FourierAttention effectively reduces KV cache memory while preserving long-context capability, its fixed-length Fourier approximation may discard part of the high-frequency temporal information in compressed dimensions. This can affect tasks that rely on exact token-level details, such as aggregation, counting, or variable tracing. Moreover, our experiments do not yet cover 70B-scale models or architectures based on Multi-head Latent Attention (MLA). Evaluating the scalability of FourierAttention on larger models and adapting it to MLA-based KV representations remain important directions for future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. U24B20181 and 62525602). We are very grateful to Ruixiao Li for his help and guidance on the Triton kernel optimization.

References

- Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*.
- Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.
- Federico Barbero, Alex Vitvitskiy, Christos Perivolaropoulos, Razvan Pascanu, and Petar Veličković. 2024. Round and round we go! what makes rotary positional encodings useful? *arXiv preprint arXiv:2410.06205*.
- Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Baobao Chang, Junjie Hu, and 1 others. 2024. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv: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. 2024. Palu: Compressing kv-cache with low-rank projection. *arXiv preprint arXiv:2407.21118*.
- OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. <https://github.com/open-compass/opencompass>.
- Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*.
- Tri Dao, Daniel Haziza, Francisco Massa, and Grigory Sizov. 2023. Flash-decoding for long-context inference.
- Haojie Duanmu, Zhihang Yuan, Xiuhong Li, Jiangfei Duan, Xingcheng Zhang, and Dahua Lin. 2024. Skvq: Sliding-window key and value cache quantization for large language models. *arXiv preprint arXiv:2405.06219*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Yao Fu. 2024. Challenges in deploying long-context transformers: A theoretical peak performance analysis. *arXiv preprint arXiv:2405.08944*.
- Yu Fu, Zefan Cai, Abdelkadir Asi, Wayne Xiong, Yue Dong, and Wen Xiao. 2024. Not all heads matter: A head-level kv cache compression method with integrated retrieval and reasoning. *arXiv preprint arXiv:2410.19258*.
- Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. 2020. Hippo: Recurrent memory with optimal polynomial projections. *Advances in Neural Information Processing Systems*, 33:1474–1487.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Siyang He, Qiqi Wang, Xiaoran Liu, Hongnan Ma, Yiwei Shi, Yuerong Song, Ying Zhu, Tianyi Liang, Zengfeng Huang, Ziwei He, and 1 others. 2026. Fouriersampler: Unlocking non-autoregressive potential in diffusion language models via frequency-guided generation. *arXiv preprint arXiv:2601.23182*.
- Ziwei He, Meng Yang, Minwei Feng, Jingcheng Yin, Xinbing Wang, Jingwen Leng, and Zhouhan Lin. 2023. Fourier transformer: Fast long range modeling by removing sequence redundancy with fft operator. *arXiv preprint arXiv:2305.15099*.
- Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. 2024. Kvquant: Towards 10 million context length llm inference

- with kv cache quantization. *arXiv preprint arXiv:2401.18079*.
- Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesht, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What’s the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.
- Dongwon Jo, Jiwon Song, Yulhwa Kim, and Jae-Joon Kim. 2025. Fastkv: Kv cache compression for fast long-context processing with token-selective propagation. *arXiv preprint arXiv:2502.01068*.
- Greg Kamradt. 2023. Needle in a haystack - pressure testing llms. https://github.com/gkamradt/LLMTest_NeedleInAHaystack.
- Mo Li, Songyang Zhang, Yunxin Liu, and Kai Chen. 2024a. Needlebench: Can llms do retrieval and reasoning in 1 million context window? *arXiv preprint arXiv:2407.11963*.
- Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024b. Snapkv: Llm knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*.
- Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, and 1 others. 2024a. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*.
- Xiaoran Liu, Ruixiao Li, Mianqiu Huang, Zhigeng Liu, Yuerong Song, Qipeng Guo, Siyang He, Qiqi Wang, Linlin Li, Qun Liu, and 1 others. 2025. Thus spake long-context large language model. *arXiv preprint arXiv:2502.17129*.
- Xiaoran Liu, Hang Yan, Chenxin An, Xipeng Qiu, and Dahua Lin. 2024b. Scaling laws of rope-based extrapolation. In *The Twelfth International Conference on Learning Representations*.
- Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024c. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*.
- Xin Men, Mingyu Xu, Bingning Wang, Qingyu Zhang, Hongyu Lin, Xianpei Han, and Weipeng Chen. 2024. Base of rope bounds context length. *arXiv preprint arXiv:2405.14591*.
- AI Meta. 2024a. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI*.
- AI Meta. 2024b. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. *Meta AI*.
- OpenAI. 2024. [O1: Openai’s first model](#). Accessed: 2024-12-25.
- Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2024. Yarn: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*.
- Qwen, :, 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, and 24 others. 2024. [Qwen2.5 technical report](#). *arXiv preprint arXiv:2412.15115*.
- Utkarsh Saxena, Gobinda Saha, Sakshi Choudhary, and Kaushik Roy. 2024. Eigen attention: Attention in low-rank space for kv cache compression. *arXiv preprint arXiv:2408.05646*.
- Dachuan Shi, Yonggan Fu, Xiangchi Yuan, Zhongzhi Yu, Haoran You, Sixu Li, Xin Dong, Jan Kautz, Pavlo Molchanov, and 1 others. 2025. Lacache: Ladder-shaped kv caching for efficient long-context modeling of large language models. *arXiv preprint arXiv:2507.14204*.
- Tianxiang Sun, Xiaotian Zhang, Zhengfu He, Peng Li, Qinyuan Cheng, Xiangyang Liu, Hang Yan, Yunfan Shao, Qiong Tang, Shiduo Zhang, Xingjian Zhao, Ke Chen, Yining Zheng, Zhejian Zhou, Ruixiao Li, Jun Zhan, Yunhua Zhou, Linyang Li, Xiaogui Yang, and 5 others. 2024. [Moss: An open conversational large language model](#). *Machine Intelligence Research*.
- Jamba Team, Barak Lenz, Alan Arazi, Amir Bergman, Avshalom Manevich, Barak Peleg, Ben Aviram, Chen Almagor, Clara Fridman, Dan Padnos, and 1 others. 2024. Jamba-1.5: Hybrid transformer-mamba models at scale. *arXiv preprint arXiv:2408.12570*.
- Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Jian Tong, Haodong Duan, Qipeng Guo, Jiaqi Wang, and 1 others. 2025. Videorope: What makes for good video rotary position embedding? *arXiv preprint arXiv:2502.05173*.
- Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. 2024a. Duoattention: Efficient long-context llm inference with retrieval and streaming heads. *arXiv preprint arXiv:2410.10819*.
- Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024b. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.
- Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and 1 others. 2024. Pyramidrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Y. X. Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. 2025. Native sparse attention: Hardware-aligned and natively trainable sparse attention. *arXiv preprint arXiv:2502.11089*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuan-dong Tian, Christopher Ré, Clark Barrett, and 1 others. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.

A Further Analysis of Dimension Heterogeneity

Regarding the observation on K cache, there exists a relation between this dimension bifurcation with RoPE. In Figure 2, we compare the attention score components of the first 70 dimensions and the last 58 dimensions of all QK states in Llama 3.1-8B and Llama 3.2-3B, as well as the effects on the downstream NIAH task after adding noise. The choice of the first 70 dimensions refers to the concept of critical dimension in Liu et al. (2024b), which corresponds to the number of dimensions with complete position information observed during the pre-training phase in LLMs with RoPE. This dimension d_{extra} can be calculated based on the head dimension d , the initial pre-training length T_{train} , and the initial rotary base β .

$$d_{\text{extra}} = 2 \left\lceil \frac{d}{2} \log_{\beta} \frac{T_{\text{train}}}{2\pi} \right\rceil$$

Some studies have found that the self-attention components before and after the critical dimension have different characteristics in long contexts (Liu et al., 2024b; Wei et al., 2025), and when the rotation angle is large, LLMs are not good at characterizing long-context features (Men et al., 2024). Therefore, we choose the critical dimension as the split and find that different dimensions before and after have different impacts on long-context tasks. For Llama3 as well as Llama 3.1 and Llama 3.2, $d = 128$, $T_{\text{train}} = 8192$, $\beta = 500000$, and the critical dimension size is 70.

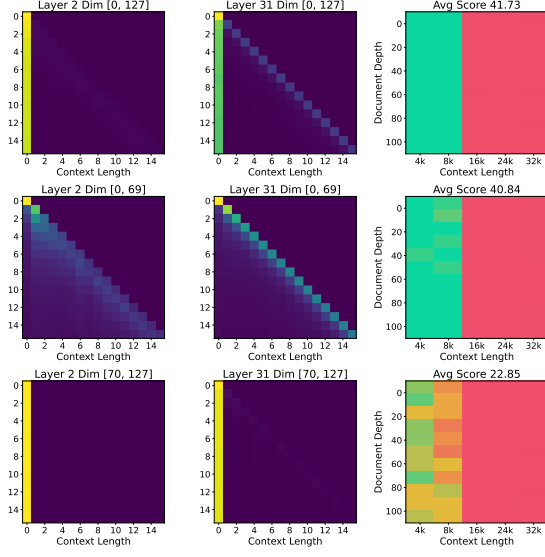
We have also observed similar conclusions on Llama 3-8B with short contexts as shown in Figure 8a. For the first 70 dimensions and the last 58 dimensions of Llama 3-8B, adding noise to the first 70 dimensions has almost no impact, while adding noise to the last 58 dimensions leads to a significant drop in performance within the pre-training length (8K). As for the case of a larger rotation angle base, we scale up the rotary base of Llama 3-8B by $13\times$ to support a 32K context. Similarly, as shown in Figure 8b, for the NTK-extrapolated Llama 3-8B, adding noise to the first 70 dimensions and the last 58 dimensions, respectively, we find that noise in the first 70 dimensions has a weaker impact, while noise in the last 58 dimensions has a greater impact.

We also validate this observation on Qwen3-8B (Yang et al., 2025), as shown in Figure 8c. For Qwen3-8B, the pre-training context length and rotary base are 4096 and 10000, respectively, resulting in a critical dimension of $d_{\text{extra}} = 92$. We therefore separately perturb the first 92 and the last 36 dimensions. Once again, perturbing the upper dimensions has a substantially greater effect on NIAH performance than perturbing the lower dimensions, demonstrating that the observation generalizes across model families and RoPE configurations.

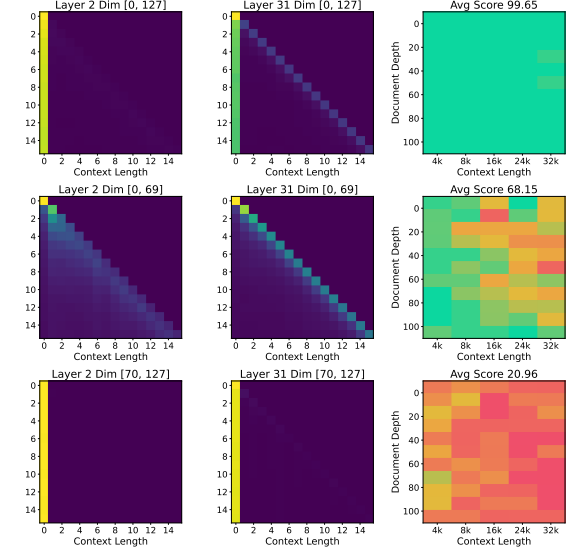
To investigate whether RoPE is responsible for this bifurcation, we further conduct experiments on Jamba 1.5-Mini (Team et al., 2024), a hybrid model whose attention layers do not use positional encoding. Because only one out of every eight layers in Jamba uses self-attention, we apply stronger Gaussian noise sampled from $\mathcal{N}(0, 16)$. As shown in Figure 8d, perturbing the first and last 64 dimensions produces degradation of similar magnitude. Moreover, the two dimension groups do not exhibit the distinct local-versus-global attention patterns observed in RoPE-based models. Across the RoPE-based models, upper dimensions contribute more strongly to global attention and attention sinks, whereas lower dimensions primarily contribute to local attention. The absence of these phenomena in Jamba further supports the connection between RoPE and the observed dimension heterogeneity.

B Mathematical Derivation

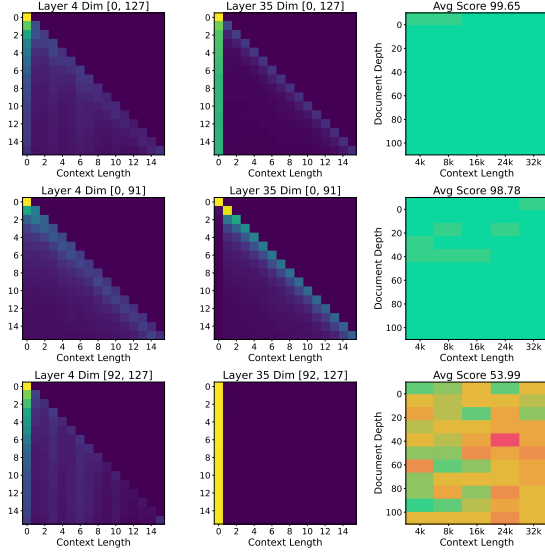
We will present the derivation process of HiPPO-FourierT in this section. Firstly, according to the HiPPO framework (Gu et al., 2020), we give the definitions of the basis functions and the measure



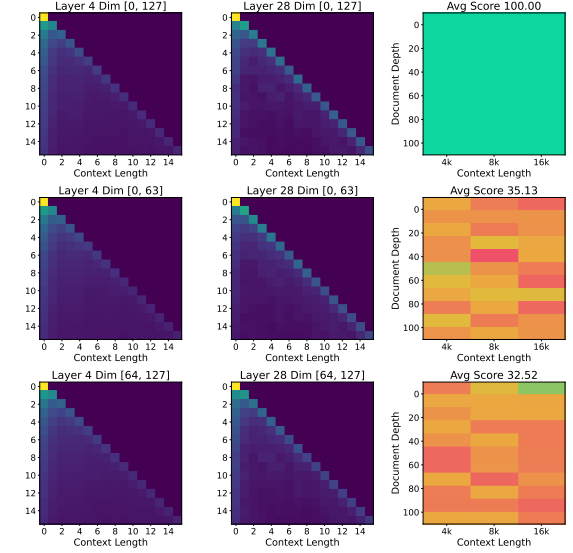
(a) Llama 3-8B



(b) Llama 3-8B with NTK scaling



(c) Qwen3-8B



(d) Jamba 1.5-Mini

Figure 8: Additional visualizations of average attention-score components and NIAH perturbation results. For the RoPE-based models in (a)–(c), lower dimensions exhibit a predominantly local attention pattern, whereas upper dimensions contribute more strongly to global attention. Accordingly, perturbing the upper dimensions causes substantially greater performance degradation than perturbing the lower dimensions. In contrast, the NoPE-based Jamba model in (d) does not exhibit a comparable dimension bifurcation, and perturbing either half of the dimensions has a similar effect.

functions. We use the Fourier bases as the basis functions $g_n(x, t)$, and adopt the translated average measure $\mu(x, t)$ within a fixed window. The formulas are as follows, where the length of the fixed window is equal to the maximum context length T supported by the model.

$$g_n(x, t) = e^{i \frac{2\pi n x}{T}}, \quad \mu(x, t) = \frac{1}{T} \mathbb{I}_{[t-T, t]} \quad (8)$$

Based on this, the input signal $f(t)$, which is the KV cache to be compressed, can be obtained as the projection $c_n(t)$ of the basis functions $g_n(x, t)$ under the measure $\mu(x, t)$.

$$\begin{aligned} c_n(t) &= \langle f_{\leq t}(x), g_n(x, t) \rangle_{\mu(x, t)} \\ &= \int_0^t f(x) \cdot g_n(x, t) \cdot \mu(x, t) dx \end{aligned} \quad (9)$$

Then, we differentiate the state $c_n(t)$.

$$\begin{aligned} \frac{dc_n(t)}{dt} &= \int_0^t f(x) \cdot \frac{\partial g_n(x, t)}{\partial t} \cdot \mu(x, t) dx + \\ &\quad \int_0^t f(x) \cdot g_n(x, t) \cdot \frac{\partial \mu(x, t)}{\partial t} dx \end{aligned} \quad (10)$$

Since $\frac{\partial g_n(x, t)}{\partial t} = 0$ and $\frac{\partial \mu(x, t)}{\partial t} = \frac{\delta_t}{T} - \frac{\delta_{t-T}}{T}$, the differentiation of $c_n(t)$ is simplified as follows.

$$\begin{aligned} \frac{dc_n(t)}{dt} &= 0 + \int_0^t f(x) \cdot g_n(x, t) \cdot \frac{\delta_t}{T} dx - 0 \\ &= \frac{f(t)}{T} \cdot e^{i\frac{2\pi nt}{T}} \end{aligned} \quad (11)$$

Considering that the actual storage of LLM is real numbers, we calculate the derivatives of the real and imaginary parts, respectively.

$$\begin{aligned} \text{Re} \left[\frac{dc_n(t)}{dt} \right] &= \frac{f(t)}{T} \cos \frac{2\pi nt}{T}, \\ \text{Im} \left[\frac{dc_n(t)}{dt} \right] &= \frac{f(t)}{T} \sin \frac{2\pi nt}{T} \end{aligned} \quad (12)$$

After discretizing them respectively, we obtain the final state update equations. Regarding the choice of discretization strategy, since the $\mathbf{A}$ matrix in HiPPO-FourierT is an identity matrix, the results obtained by different discretization methods in HiPPO-FourierT are only different in step size. We choose the simplest forward Euler discretization as follows.

$$c_{t+1}^{(n)} = \begin{cases} c_t^{(n)} + \frac{f_t}{T} \cos \frac{2\pi nt}{T} & n = 2m \\ c_t^{(n)} + \frac{f_t}{T} \sin \frac{2\pi nt}{T} & n = 2m + 1 \end{cases} \quad (13)$$

$m = 0, 1, \dots, N-1$

Then we can derive the compression, update, and decompression functions as shown in Equation 5 in our paper,

$$\begin{aligned} c_{t+1} &= \frac{1}{T} \mathcal{F}_{N \times t} f_t \\ \mathcal{F}_{k \times t} &= \begin{bmatrix} 1 & 1 & \dots & 1 \\ 0 & 0 & \dots & 0 \\ 1 & \cos \frac{2\pi}{T} & \dots & \cos \frac{2\pi t}{T} \\ 0 & \sin \frac{2\pi}{T} & \dots & \sin \frac{2\pi t}{T} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \cos \frac{2\pi(N-1)}{T} & \dots & \cos \frac{2\pi(N-1)t}{T} \\ 0 & \sin \frac{2\pi(N-1)}{T} & \dots & \sin \frac{2\pi(N-1)t}{T} \end{bmatrix}, \end{aligned} \quad (14)$$

where

$$\begin{aligned} c_{t+1} &= [c_{t+1}^{(0)}, c_{t+1}^{(1)}, \dots, c_{t+1}^{(2k-2)}, c_{t+1}^{(2k-1)}]^\top \\ f_t &= [f_0, f_1, \dots, f_t]^\top \end{aligned}$$

Regarding discretization, since the $\mathbf{A}$ matrix in HiPPO-FourierT is a zero matrix, the results obtained by different discretization methods in HiPPO-FourierT are only different in step size. We finally chose the simplest forward Euler discretization. In the comparative experiments, we have compared HiPPO-LegT with the bilinear discretization method, which is the best-performing discretization method reported in Gu et al. (2020).

C Additional Experiment Results

In this section, we provide supplementary results on decoding efficiency and generalization across head dimensions, model scales, and parallel deployment settings, followed by detailed NIAH visualizations.

To demonstrate the effectiveness of our method with different head dimension sizes (head_dim), we also conduct the same experiments on the Llama 3.2-1B model (Meta, 2024a), which has a head dimension of 64, a significant difference from the 128 of the Llama 3.2-3B model. The results are in Table 6. The result shows that our FourierAttention achieves consistent superiority over other methods, and shows the closest performance with LLMs with full attention. This strongly proves that our method possesses good cross-architecture generalization ability.

Beyond different head dimensions, we further evaluate the scalability of FourierAttention to larger models and distributed inference. Specifically, we extend FourierAttention to Qwen2.5-14B-Instruct-1M and run the evaluation with tensor parallelism across two H200 GPUs. As shown in Table 6, FourierAttention achieves the best average Long-Bench performance among the compressed methods and remains closest to the full-KV baseline. These results demonstrate that FourierAttention generalizes beyond the smaller models evaluated in the main experiments and can be naturally deployed in a multi-GPU tensor-parallel environment.

Figure 9 presents the detailed “Needle-In-A-Haystack” (NIAH) visualization for Llama 3.1-8B. Consistent with our main results, FourierAttention achieves performance closest to the original pre-trained model across the 32K context length. While

Table 6: Additional LongBench results on Llama 3.2-1B with a head dimension of 64 and Qwen2.5-14B-Instruct-1M under tensor parallelism across two H200 GPUs. Full-KV results are shown in gray, and the best average results among cache-optimized methods are bolded.

	Single-Doc			Multi-Doc			Summary			Few-shot			Synthetic		Code		Avg.
	NQ	Qsp	MF	HQ	WQ	Msq	GR	QS	MN	TR	TQ	SS	PC	PR	LCC	Re-P	
Llama 3.2-1B	5.1	20.2	24.9	9.1	10.3	6.2	28.2	20.8	23.4	67.5	82.1	39.5	1.4	3.4	56.7	57.2	33.1
+ SLM	5.8	11.8	16.1	6.1	9.5	3.4	16.5	19.8	20.1	50.5	68.4	37.2	1.3	4.1	48.1	48.4	27.1
+ SnapKV	4.1	19.7	24.2	9.3	10.1	6.0	28.4	20.4	23.2	66.5	81.2	39.2	1.5	3.4	46.2	47.7	30.2
+ PyramidKV	4.3	19.6	24.3	9.1	10.1	6.0	28.4	20.5	23.1	67.0	81.3	39.3	1.5	3.4	46.2	47.7	30.2
+ FA (ours)	2.4	17.1	24.0	8.8	10.5	4.9	16.2	19.6	19.6	63.5	80.8	39.5	1.3	4.1	56.8	51.7	30.8
Qwen2.5-14B-Inst.-1M	22.3	46.5	49.0	59.1	61.0	36.1	33.0	24.1	22.5	81.0	89.0	48.3	16.1	99.0	68.8	67.9	54.2
+ SLM	17.3	47.3	44.6	51.5	60.4	26.3	35.6	23.9	25.2	80.0	87.8	46.5	10.5	71.7	58.8	44.2	46.7
+ SnapKV	24.6	46.8	47.2	55.9	60.7	33.9	35.7	24.8	25.3	80.8	88.4	47.8	15.5	98.3	59.4	61.0	52.0
+ PyramidKV	25.4	46.8	47.2	56.1	60.7	33.9	35.4	24.8	25.3	80.8	88.9	47.8	15.5	98.3	59.4	60.9	52.1
+ FA (ours)	21.8	47.9	48.5	56.1	58.8	32.7	31.6	25.0	25.5	80.5	87.6	47.2	16.2	99.0	68.3	63.1	53.0

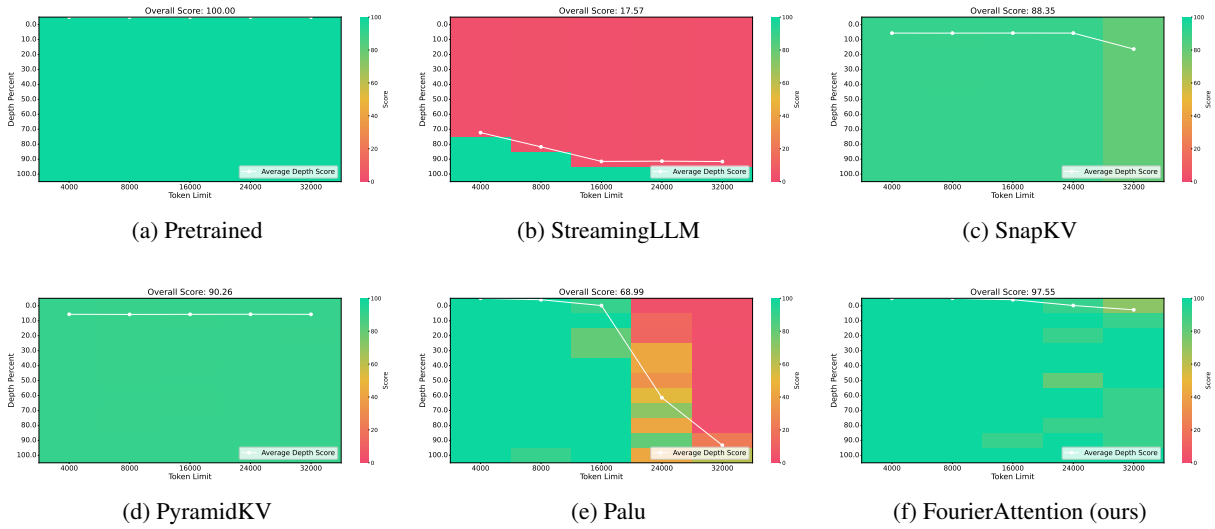


Figure 9: Results of Llama 3.1-8B (Dubey et al., 2024) on Needle-In-A-Haystack (Kamradt, 2023). FourierAttention achieves the highest average score over StreamingLLM, SnapKV, PyramidKV, and Palu, and shows the closest performance with LLMs with full attention.

Table 7: Decoding throughput (tokens/s) for SnapKV, Palu, and FourierAttention on the Llama series. OOM denotes out-of-memory.

Method	8K	16K	32K	48K	64K	80K
Llama 3.1-8B Decoding Throughput (tokens/s)						
SnapKV	77.5	40.4	19.7	16.5	OOM	OOM
Palu	109.5	50.9	27.6	17.3	13.7	OOM
FA (ours)	127.1	64.6	35.0	21.2	17.5	11.8
Llama 3.2-3B Decoding Throughput (tokens/s)						
SnapKV	118.3	60.9	30.5	23.6	OOM	OOM
Palu	152.8	70.7	39.4	30.0	18.7	14.9
FA (ours)	185.2	93.1	55.3	33.7	24.2	20.7

methods like SnapKV and PyramidKV are theoretically suitable for retrieval tasks, they still exhibit visible recall errors. Furthermore, although Palu maintains stable approximation under mod-

erate compression, it suffers from significant performance degradation at higher compression rates (e.g., $\sim 75\%$) due to insufficient granular analysis of KV cache features. In contrast, by identifying and preserving critical KV dimensions while efficiently compressing the others, FourierAttention maximally retains long-context capabilities and demonstrates consistent superiority over other cache optimization baselines.

Table 7 reports decoding throughput across different context lengths. FourierAttention maintains competitive throughput on both models and generally performs better than SnapKV and Palu in the evaluated settings. It also supports longer contexts where some baselines encounter OOM. These results suggest that the block-wise Fourier reconstruction in FlashFourierAttention introduces limited

decoding overhead while preserving the memory benefits of FourierAttention.

C.1 Effective KV-Cache Budget

We calculate the effective compression ratio by accounting for both the dimensions retained in their original temporal form and those represented by fixed-length Fourier states. Let ρ denote the overall proportion of compressed KV dimensions and $L_{\text{middle}} = L - L_{\text{init}} - L_{\text{local}}$. The effective retention and compression ratios are

$$r_{\text{ret}} = \frac{L_{\text{init}} + L_{\text{local}}}{L} + \frac{(1 - \rho)L_{\text{middle}} + \rho N'}{L}, \quad (15)$$

$$r_{\text{comp}} = 1 - r_{\text{ret}}.$$

Under our default setting, $L = 32000$, $L_{\text{init}} = 4$, $L_{\text{local}} = 1024$, $N' = 1024$, and $\rho = 0.76$. Thus, $L_{\text{middle}} = 30972$, and

$$r_{\text{comp}} = 1 - \frac{4 + 1024 + 0.24 \times 30972}{32000} - \frac{0.76 \times 1024}{32000} \approx 71.1\%.$$

Therefore, FourierAttention retains an effective KV-cache budget of approximately 28.9%, which is used to configure the baselines in our experiments.

D Analysis of Method Design Choices

We examine three design choices of FourierAttention: the layer-wise and K/V compression configuration, the numbers of initial and recent tokens, and the basis function used for temporal compression. These experiments evaluate the robustness of FourierAttention and justify its default configuration.

D.1 Robustness to Compression Configurations

We evaluate configuration robustness on Llama 3.2-3B and Qwen2.5-3B. The variants shift the boundary between low and high layers by $\pm 7\%$ (dis1/dis2), inversely adjust the K/V compression ratios by $\pm 5\%$ (dis3/dis4), apply a slight perturbation (dis5), or use uniform compression across layers and K/V caches (uniform). The number of compressed dimensions, thus the effective KV-cache budget, is fixed across all variants. Across

both models, all variants match or outperform PyramidKV in average score, while the original configuration achieves the best overall performance. These results demonstrate that FourierAttention is robust to moderate configuration changes while supporting our asymmetric allocation.

D.2 Ablation of Initial Local Tokens

Following StreamingLLM (Xiao et al., 2024b), we use $L_{\text{init}} = 4$ to preserve attention sinks and $L_{\text{local}} = 1024$ to retain recent local and high-frequency information. Meanwhile, we examine the sensitivity to these choices by varying $L_{\text{init}} \in \{2, 4, 8\}$ with $L_{\text{local}} = 1024$, and $L_{\text{local}} \in \{512, 1024, 2048\}$ with $L_{\text{init}} = 4$. As shown in Table 10, FourierAttention remains stable across different numbers of initial tokens and consistently outperforms the baselines under all tested local-window sizes. The default setting achieves the best overall performance while avoiding a larger local window, providing a favorable accuracy–memory trade-off.

D.3 Choice of Basis Function

We further ablate the choice of basis function used for temporal compression. Although FourierAttention employs HiPPO-FourierT for compression, Gu et al. (2020) proposes and claims polynomial basis functions like LegT generally offer superior performance. To verify this in our context, we compare LegT and FourierT in reconstructing KV caches from LLMs while maintaining identical sliding window sizes. As illustrated in Figure 10, we evaluate their reconstruction effects on 4 randomly selected KV cache dimensions from layer 0 of Llama 3.2. Under equivalent state dimensions¹, FourierT consistently achieves lower reconstruction loss than LegT. This empirical evidence supports our architectural decision to utilize FourierT for effective KV cache compression.

E Comparison with Head-Level Methods

Although FourierAttention is a dimensionally heterogeneous approach, the uneven distribution of long-context-sensitive dimensions across attention heads intrinsically reflects heterogeneity at head level. To investigate this phenomenon, we compare FourierAttention with DuoAttention (Xiao et al., 2024a) and HeadKV (Fu et al., 2024), both

¹FourierT uses lower-order basis functions since FourierAttention’s state size is twice the number of states.

Table 8: LongBench ablation results on Llama 3.2-3B across subtasks for different configuration variants.

	Single-Doc			Multi-Doc			Summary			Few-shot			Synthetic		Code		Avg.
	NQ	Qsp	MF	HQ	WQ	Msq	GR	QS	MN	TR	TQ	SS	PC	PR	LCC	Re-P	
Llama-3.2-3B	10.3	21.7	35.5	9.6	12.8	6.8	30.2	23.8	28.2	70.0	87.2	38.2	0.0	7.0	70.0	66.4	38.0
+ PyramidKV	8.9	21.4	35.7	9.5	12.8	6.8	30.4	23.6	28.2	69.5	86.9	38.7	0.0	6.8	58.7	57.3	35.1
+ FA(ori)	12.8	21.1	35.8	9.8	11.6	6.5	23.0	23.8	24.1	69.0	87.0	39.4	0.0	6.2	69.8	63.3	37.0
+ FA(uniform)	6.9	19.8	34.0	10.5	12.9	6.4	18.9	20.9	25.2	69.0	86.4	37.5	0.1	5.2	69.8	62.1	36.0
+ FA(dis1)	5.8	21.0	35.0	10.1	13.1	7.3	21.6	22.4	24.3	69.5	86.5	37.8	0.0	5.6	69.7	62.2	36.3
+ FA(dis2)	5.1	20.7	34.9	10.7	13.3	6.7	22.0	23.2	24.3	69.5	87.0	37.9	0.0	4.8	70.0	62.3	36.4
+ FA(dis3)	6.2	20.1	34.7	10.5	12.9	7.4	19.7	22.2	24.5	70.0	86.7	37.6	0.0	5.1	69.5	61.3	36.0
+ FA(dis4)	4.7	20.2	35.5	10.6	12.7	6.5	22.5	23.8	24.9	69.5	87.1	37.8	0.0	5.5	69.9	62.6	36.5
+ FA(dis5)	4.7	20.5	35.6	10.5	13.4	6.6	22.5	23.7	25.4	69.5	86.7	37.8	0.0	5.3	69.9	62.9	36.5

Table 9: LongBench ablation results on Qwen2.5-3B across subtasks for different configuration variants.

Variant	Single-Doc			Multi-Doc			Summary			Few-shot			Synthetic		Code		Avg.
	NQ	Qsp	MF	HQ	WQ	Msq	GR	QS	MN	TR	TQ	SS	PC	PR	LCC	Re-P	
Qwen2.5-3B	14.3	29.9	45.9	38.4	33.5	20.6	29.7	24.2	25.2	69.0	89.5	45.3	2.0	8.0	69.3	65.6	42.7
+ PyramidKV	14.5	30.0	45.6	37.1	33.5	19.7	29.4	23.8	25.0	69.0	88.5	45.0	1.5	8.0	57.4	55.2	39.5
+ FA(ori)	6.4	27.7	43.3	30.4	31.7	16.3	22.0	23.4	24.7	66.5	88.9	45.2	2.0	7.5	68.9	62.0	40.1
+ FA(uniform)	7.4	25.1	41.7	30.8	30.1	14.5	21.2	23.0	25.0	65.5	88.9	45.9	2.5	7.5	68.4	60.8	39.5
+ FA(dis1)	7.3	25.9	41.4	32.0	31.1	14.4	21.4	22.6	24.6	64.5	88.9	44.5	2.0	6.5	68.1	61.7	39.5
+ FA(dis2)	7.5	26.0	42.5	32.8	30.8	15.2	21.4	22.6	24.6	64.5	88.0	44.1	2.0	7.0	68.5	60.7	39.5
+ FA(dis3)	5.9	27.4	41.0	32.1	30.8	17.7	21.2	22.8	24.5	62.5	88.7	44.6	2.0	7.0	68.2	61.8	39.6
+ FA(dis4)	5.5	27.2	42.5	30.6	30.8	17.0	22.0	23.8	24.0	64.0	89.6	44.4	2.0	7.5	67.7	61.3	39.6
+ FA(dis5)	6.8	26.9	41.5	29.3	30.6	17.0	21.7	23.0	24.6	64.5	88.7	44.5	2.0	7.0	68.2	61.3	39.5

Table 10: LongBench average scores on Llama 3.2-3B with different numbers of initial tokens L_{init} and local-window sizes L_{local} . The best result in each column is bolded.

Method	Initial Tokens ($L_{local} = 1024$)			Local Window ($L_{init} = 4$)		
	$L_{init} = 2$	$L_{init} = 4$	$L_{init} = 8$	$L_{local} = 512$	$L_{local} = 1024$	$L_{local} = 2048$
StreamingLLM	31.20	31.30	31.26	29.47	31.30	32.64
SnapKV	35.15	35.16	35.16	35.14	35.16	35.18
PyramidKV	35.15	35.15	35.16	35.16	35.15	35.19
FA (ours)	36.76	37.00	36.74	36.37	37.00	35.66

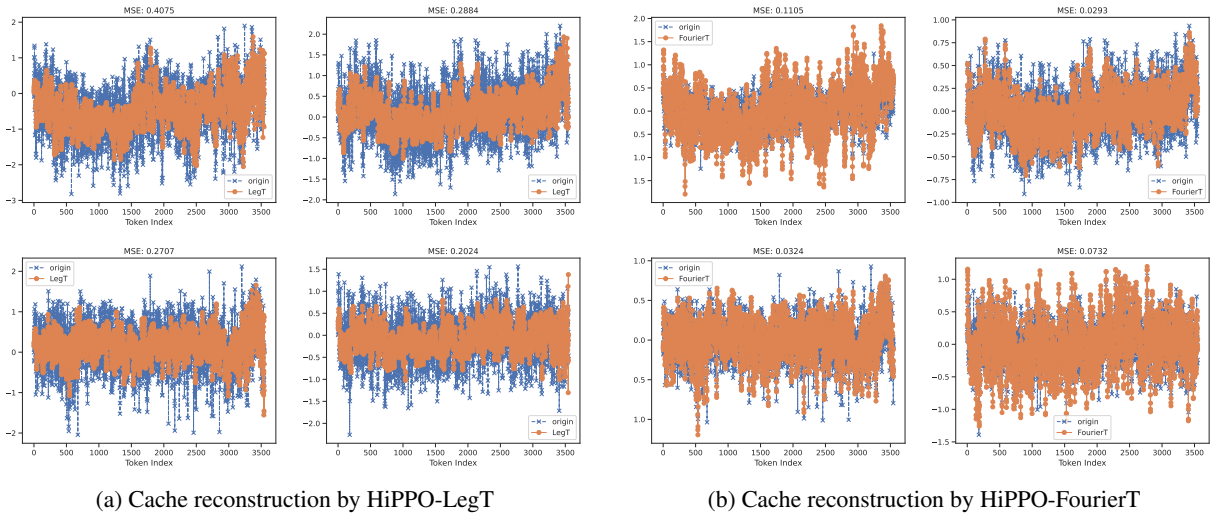


Figure 10: Visualization of KV cache reconstruction in Llama 3.2-3B for different basis functions, LegT and FourierT under HiPPO framework (Gu et al., 2020). FourierT outperforms LegT in cache reconstruction with lower reconstruction loss.

Table 11: Jaccard similarity of important heads on FourierAttention(Ours), DuoAttention, and HeadKV across different models.

Model	C vs. A	C vs. B	A vs. B
Llama 3.1-8B-Instruct	0.762	0.755	0.736
Llama 3-8B-Instruct	0.713	0.725	0.723
Mistral-7B-Instruct-v0.2	0.884	0.875	0.889

of which identify "Retrieval Heads" specialized in long-context information and reasoning. We first define three sets of heads for each method:

- Set A (DuoAttention): Based on Passkey Retrieval attention scores in DuoAttention, we select heads exceeding a threshold γ and define the set size as K .
- Set B (HeadKV): We select the Top- K heads based on the importance scores provided by HeadKV.
- Set C (Ours): We count the number of "long-context-sensitive dimensions" (summing K and V) identified by our method within each head, and select the Top- K heads with the highest counts.

We conduct experiments on Llama 3.1-8B-Instruct, Llama 3-8B-Instruct, and Mistral-7B-Instruct-v0.2 used in [Xiao et al. \(2024a\)](#); [Fu et al. \(2024\)](#). We employ the Jaccard similarity coefficient to measure the overlap between these sets. As shown in Table 11, our method exhibits high consistency with DuoAttention and HeadKV, demonstrating that the important heads derived from our dimensional analysis largely overlap with the "Retrieval Heads" identified in prior work.

To further demonstrate the superiority of our fine-grained "dimension-level" approach over coarse-grained "head-level" strategies, we design a comparative experiment named "HeadFA." For each layer, we fully retain the Top- t heads richest in sensitive dimensions, where $t = \lceil \frac{\text{sensitive dims}}{\text{total dims}} \times \text{total heads} \rceil$, and apply FA compression to the rest.

The results in Table 4 show that HeadFA underperforms FourierAttention. This reveals a deeper mechanism: the functional "importance" of an attention head fundamentally stems from its concentration of "long-context-sensitive dimensions." This finding provides a perspective that is more fundamental and fine-grained than the "head" level for understanding the intrinsic principles of attention mechanisms.

F Integration with Low-Bit Quantization

FourierAttention and KIVI reduce KV-cache storage along complementary axes. FourierAttention reduces temporal redundancy by representing selected KV dimensions with fixed-length Fourier states, while KIVI reduces numerical precision through asymmetric low-bit quantization. To combine them, we retain the original dimension selection and Fourier transformation of FourierAttention, and apply KIVI 4-bit quantization to the conventional token-wise KV entries that remain explicitly stored by FourierAttention. Following KIVI ([Liu et al., 2024c](#)), the K cache is quantized per channel, whereas the V cache is quantized per token. During decoding, the quantized token-wise cache is dequantized on demand, while the Fourier-compressed dimensions are reconstructed through the original FourierAttention path. The two components are then jointly used for attention computation. This composition requires neither additional training nor modification of the dimension-selection procedure.

We evaluate the combined method on Llama 3.2-3B under the same LongBench settings and compare it with the full KV cache, full-precision FourierAttention, and KIVI 4-bit quantization. As shown in Table 12, FA+KIVI achieves an average score of 36.8, closely matching FourierAttention (37.0) and KIVI (36.8), compared with 38.0 for the full KV cache. The combined method also remains competitive across individual tasks and outperforms both standalone methods on several datasets, including HotpotQA, 2WikiMQA, and SAMSum. These results indicate that the quantization error does not compound substantially with Fourier reconstruction, supporting the compatibility between Fourier compression and low-bit KV-cache quantization.

G Triton Implementation Details

Regarding Triton implementation details, since PyTorch itself already provides efficient support for Fourier transforms, we only need to design and integrate a custom kernel for the decoding stage, focusing on solving the decompression calculations involved in the decoding stage, and wrapping the decompression within the attention calculations to avoid the read-write of full-size KV cache. In this process, we applied three techniques: FlashDecoding, dimension reordering, and compression delay.

FlashDecoding is a fundamental method for de-

Table 12: LongBench results on Llama 3.2-3B with the full KV cache, full-precision FourierAttention (FA), KIVI 4-bit quantization, and their combination. The best and second-best results among compressed methods are bolded and underlined, respectively.

	Single-Doc			Multi-Doc			Summary			Few-shot			Synthetic		Code		Avg.
	NQ	Qsp	MF	HQ	WQ	Msq	GR	QS	MN	TR	TQ	SS	PC	PR	LCC	Re-P	
Llama 3.2-3B	10.3	21.7	35.5	9.6	12.8	6.8	30.2	23.8	28.2	70.0	87.2	38.2	0.0	7.0	70.0	66.4	38.0
+ FA (ours)	12.8	<u>21.1</u>	<u>35.8</u>	<u>9.8</u>	11.6	6.5	<u>23.0</u>	23.8	24.1	69.0	87.0	<u>39.4</u>	0.0	<u>6.2</u>	69.8	<u>63.3</u>	37.0
+ KIVI-4bit	<u>12.5</u>	22.1	36.2	<u>9.1</u>	<u>11.9</u>	7.0	25.1	22.8	24.3	69.0	86.5	38.6	0.0	7.6	66.9	63.7	<u>36.8</u>
+ FA+KIVI	12.0	20.4	<u>35.8</u>	10.3	12.2	<u>6.8</u>	22.9	<u>23.2</u>	<u>24.2</u>	69.0	<u>86.9</u>	40.9	0.0	5.7	<u>68.6</u>	62.7	<u>36.8</u>

coding acceleration (Dao et al., 2023). Building on the iterative calculation of self-attention in FlashAttention, it targets the parallel computation of attention in the decoding phase. During decoding, it first segments and computes in parallel, and within each segment, it calculates the attention numerators, denominators, and max attention scores in a block-cycling manner. Finally, it merges the results from different segments. Since the inverse Fourier transform is essentially a matrix multiplication that can be split and computed in parallel along the sequential dimension, the decompression logic of FourierAttention can naturally be adapted to FlashDecoding. Each segment loads the complete compressed state, first decompresses the K cache within the corresponding block, concatenates it with the retained part, multiplies QK, and then gets the attention distribution. After that, it decompresses the corresponding V cache, concatenates it with the retained part, and outputs the attention result within the block.

Secondly, the dimension indices of the KV cache that need to be compressed and those that need to be fully retained, denoted as $\mathcal{D}^{ku}$, $\mathcal{D}^{kc}$, $\mathcal{D}^{vu}$, $\mathcal{D}^{vc}$, are theoretically discrete. However, to improve the loading speed, following the approach in Duanmu et al. (2024), we reorder the dimensions of the QK and VO matrices according to these indices. Specifically, the dimensions in $\mathcal{D}^{ku}$, $\mathcal{D}^{vu}$ that do not need to be compressed are placed at the lower part of the QK features and VO features of each head, respectively. Conversely, the dimensions in $\mathcal{D}^{kc}$, $\mathcal{D}^{vc}$ that need to be compressed are placed at the upper part of the QK features and VO features of each head. In addition, we also record the average value and standard deviation of each dimension in the original and reconstructed cache, and use these to standardize, thus maintaining the numeric stability.

Finally, during the decoding process, theoretically, we need to continuously compress the KV cache that exceeds the local length into an interme-

diate state. However, if the compression process is called every time during decoding, it would lead to a significant waste of computation. Therefore, we set a delayed compression decoding step, and only compress the exceeded local part of the KV cache once after decoding for several steps, reducing the impact of compression on latency.