Resonating with RoPE: Spectral Quantization for High-Fidelity Key Cache Compression

Xuefei Wang[1], Haoyu Tang[1]*, Tianyuan Liang[1], Zhibin Wang[1], Yupeng Hu[1], Weili Guan[2]

[1]Shandong University, [2]Harbin Institute of Technology (Shenzhen)
xfwang@mail.sdu.edu.cn, tanghao258@sdu.edu.cn, liangtianyuan@mail.sdu.edu.cn,
wang_z_b@mail.sdu.edu.cn, huyupeng@sdu.edu.cn, guanweili@hit.edu.cn

Abstract

The linear growth of KV cache bottlenecks long-context LLMs, yet RoPE-induced oscillations complicate Key cache quantization. To address this issue, we propose SpectrumQuant, a frequency-domain framework that utilizes the Discrete Cosine Transform (DCT) to convert these oscillations into sparse spectral representations. Specifically, our pipeline integrates dominant frequency extraction, hybrid bit-width allocation, and high-frequency preemphasis to maximize fidelity while minimizing memory footprint. To eliminate computational overhead, we develop fused Triton kernels featuring deferred inverse transformation and on-chip sparse accumulation. Extensive experiments on several benchmarks confirm SpectrumQuant achieves efficient compression with performance and latency comparable to FP16 baselines.

1 Introduction

As Large Language Models (LLMs) evolve toward ultra-long contexts (e.g., 1M tokens) (Achiam et al., 2023; Grattafiori et al., 2024; Yang et al., 2025), the KV (Key-Value) cache, essential for reusing historical states to avoid redundant computation, exhibits linear growth. Consequently, its memory footprint rapidly eclipses that of model weights, emerging as the primary bottleneck constraining long-text inference (Kwon et al., 2023).

To alleviate this memory crisis, the KV cache quantization, which aims to represent key and value tensors in low-precision formats, has emerged as a promising direction for reducing inference memory cost. Despite great efforts dedicated to this area (Liu et al., 2024b; He et al., 2024; Wu et al., 2025), a major challenge remains unresolved: **the RoPE dilemma**. As the most popular positional encoding scheme in state-of-the-art LLMs, RoPE introduces position-dependent rotations that mix paired channels, which inevitably disrupts the channel-wise magnitude consistency of key vectors (Hooper et al., 2024). This issue renders the compression of key caches substantially more challenging than that of value caches.

This raises a critical question: How exactly does RoPE affect the distribution of key vectors? To investigate, we visualized the post-RoPE key vector along different dimensions. It can be observed that RoPE induces highly structured, periodic oscillations in the token dimension (see Figure 1, top-left). This pattern suggests a natural solution: mapping the key vector to the frequency domain. We employ the Discrete Cosine Transform (DCT)—which utilizes cosine basis functions structurally isomorphic to RoPE's rotational mechanism—to analyze the signal. As shown in Figure 1 (bottom-left), several critical observations emerge: (1) **Energy Concentration**: The majority of spectral energy is concentrated in a narrow spectral region (a few adjacent coefficients); (2) **Dynamic Range Reduction**: Apart from this region, the residual signal exhibits minimal fluctuation; and (3) **Structural Consistency**: The location of this region remains largely consistent across different layers and heads (See Figure 4 in Appendix A).

These observations motivate the core philosophy of our approach: to represent the dominant spectral peak with high precision while encoding residual frequency-domain signals with lower bit-widths. Guided by this principle, we introduce **SpectrumQuant**, a key cache quantization framework in the frequency domain to neutralize RoPE-induced fluctuations. Specifically, **SpectrumQuant** utilizes DCT to transform these oscillations into a highly sparse spectral representation. We then employ a Dominant Frequency Extraction module to isolate the energy-concentrating peak for high-precision encoding. For the remaining residual signals, recognizing that low and high frequencies contribute unequally to reconstruction,

*Corresponding author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 37328–37348

the Hybrid Bit-width Allocation mechanism is proposed, which assigns heterogeneous bit-widths to distinct frequency bands according to their importance, thereby balancing fidelity with compression ratios. Moreover, we introduce High-Frequency Pre-emphasis to align dynamic ranges across bands, allowing for shared quantization to further reduce the memory footprint. Finally, to mitigate the computational cost of frequency-domain processing, we develop deeply customized fused kernels using Triton (Tillet et al., 2019). By maximizing operator fusion for memory-intensive steps and leveraging SRAM for end-to-end **On-Chip Computation**, SpectrumQuant addresses the latency typically incurred by spectral transformations, achieving efficient long-context inference with negligible overhead.

The main contributions are summarized as follows:

- We revisit key cache quantization from a spectral analysis perspective, revealing that RoPE-induced oscillations translate into highly concentrated energy distributions in the frequency domain. Based on this insight, we propose SpectrumQuant, which utilizes time-dimension DCT to effectively capture and compress these semantic signals.
- We achieve high-fidelity and low-latency inference by integrating Dominant Frequency Extraction, Hybrid Bit-width Allocation, and High-Frequency Pre-emphasis with high-performance fused Triton kernels that resolve memory bandwidth bottlenecks through on-chip computation.
- Extensive experiments on diverse benchmarks demonstrate that SpectrumQuant achieves efficient compression while maintaining performance comparable to FP16 baselines, exhibiting consistent robustness across models of varying architectures and sizes.

2 Background

2.1 RoPE and KV Cache

Consider the l-th layer of a Transformer architecture. Let the input hidden states be denoted as $\mathbf{X} \in \mathbb{R}^{T \times D}$. For any given attention head, the Query, Key, and Value vectors are projected via linear transformations:

$$\mathbf{Q}, \mathbf{K}, \mathbf{V} = \mathbf{X}\mathbf{W}_Q,\ \mathbf{X}\mathbf{W}_K,\ \mathbf{X}\mathbf{W}_V.$$

To incorporate positional information, modern large language models (LLMs) widely adopt Rotary Positional Embeddings (Su et al., 2023), which rotate the key vector $\mathbf{K}$ by pairing its channels in a 2D plane based on their position t. For the c-th channel pair (corresponding to dimensions $2c$ and $2c+1$), the transformation at position t is defined as:

$$\begin{pmatrix} \tilde{k}_{t,2c} \\ \tilde{k}_{t,2c+1} \end{pmatrix} = \begin{pmatrix} \cos(t\theta_c) & -\sin(t\theta_c) \\ \sin(t\theta_c) & \cos(t\theta_c) \end{pmatrix} \begin{pmatrix} k_{t,2c} \\ k_{t,2c+1} \end{pmatrix}, \tag{1}$$

where the frequencies $\theta_c = b^{-2c/d}$ decay exponentially across channel pairs c (with b typically set to a large number like 10000).

To accelerate inference, the KV cache stores historical key and value vectors ($\tilde{\mathbf{K}}_{<t}, \mathbf{V}_{<t}$). However, its GPU memory footprint grows linearly with sequence length T and, in long-context scenarios ($T \geq 128K$), can exceed the model weights, becoming the primary bottleneck for throughput and context extension.

2.2 Quantization

Quantization reduces memory usage by mapping floating-point values to low-bit integers. A common approach is the standard asymmetric uniform quantization. For a given floating-point vector $\mathbf{X}$, the quantization process first computes the group-wise scaling factor s and zero-point z, and maps the values to b-bit integers $\mathbf{X}_q$:

$$\begin{aligned} s &= \frac{\max(\mathbf{X}) - \min(\mathbf{X})}{2^b - 1}, \quad z = \min(\mathbf{X}), \\ \mathbf{X}_q &= \text{Clamp}\left(\left\lfloor \frac{\mathbf{X} - z}{s} \right\rceil, 0, 2^b - 1 \right), \end{aligned} \tag{2}$$

The dequantization process then restores the integers to approximate values via $\hat{\mathbf{X}} = s \cdot \mathbf{X}_q + z$. This procedure inevitably introduces quantization error, and under the assumption of a uniform error distribution, the expected mean squared error is theoretically bounded by the scaling factor s (Peters et al., 2023):

$$\mathbb{E}\left[(\hat{\mathbf{X}} - \mathbf{X})^2 \right] = \frac{s^2}{12}. \tag{3}$$

However, RoPE poses a critical challenge for key cache quantization. It introduces high-amplitude periodic oscillations that drastically expand the dynamic range ($\max(\mathbf{X}) - \min(\mathbf{X})$). This forces a large scaling factor s, creating a coarse quantization grid. Consequently, subtle semantic details

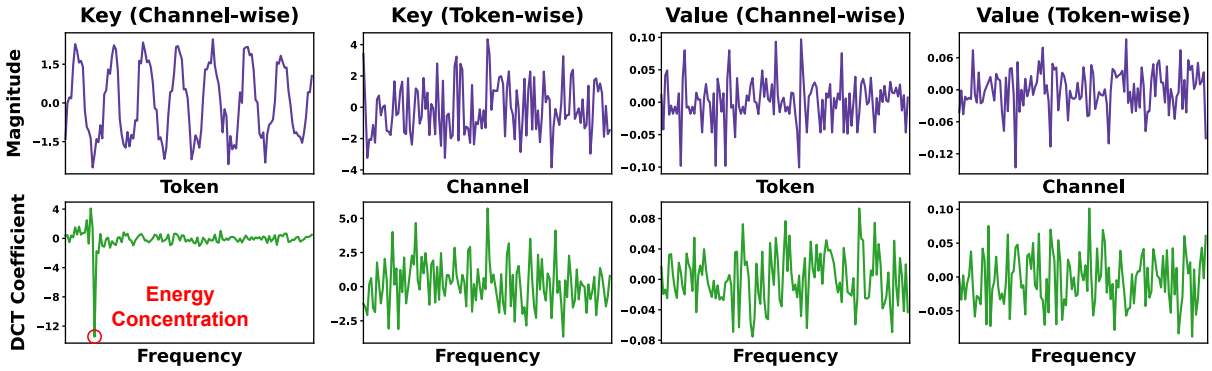

Figure 1: Visualization of original magnitudes (top) and the corresponding DCT coefficients (bottom). Significant energy concentration is observed exclusively in **the key cache along the token dimension** (leftmost).

are drowned out by quantization noise, leading to severe performance degradation in low-bit settings.

2.3 Related Works

KV Cache Quantization Recent studies have explored multiple quantization strategies for KV cache compression in LLMs. ZipCache (He et al., 2024) employs channel-separable token-wise quantization, whereas KIVI (Liu et al., 2024b) adopts channel-wise quantization for Keys and token-wise for Values. To further handle outliers and preserve fidelity, SpinQuant (Liu et al., 2024a) and PolarQuant (Wu et al., 2025) introduce coordinate transformations, such as Hadamard transform or polar transformation, before quantization. Additionally, KVQuant (Hooper et al., 2024) proposes quantizing at the pre-RoPE stage to mitigate the quantization error introduced by RoPE.

Frequency Domain Method in Large Language Models Frequency domain analysis provides a unique perspective for optimizing LLM efficiency. Existing works have successfully applied spectral methods to diverse tasks, including accelerating token mixing (Lee-Thorp et al., 2022), reducing sequence redundancy (He et al., 2023; Kai et al., 2025), and minimizing storage in fine-tuning (Gao et al., 2024b). Furthermore, frequency-based approaches have been leveraged for model weight and activation quantization (Chen et al., 2025; Zhao et al., 2026) and theoretical analysis of positional encodings (Ruscio et al., 2025). While SpecQuant (Zhao et al., 2026) similarly employs spectral transformations, it relies on weight smoothness for weight and activation quantization. In contrast, our method leverages the spectral sparsity induced by RoPE, targeting the dynamic KV cache without any offline calibration.

3 Method

In this section, we present SpectrumQuant, a spectrum-based quantization framework for key caches. We first elucidate the spectral characteristics of the key cache through empirical observations (§3.1) and theoretical analysis (§3.2). Subsequently, we detail the proposed quantization scheme (§3.3) and the hardware-optimized implementation (§3.4).

3.1 Motivation

To explore potential sparsity within the KV cache, we introduce the Discrete Cosine Transform (DCT) as our analytical tool (see Appendix B for details). Based on spectral analysis across different dimensions, we derive three key observations:

Observation 1: Energy Concentration in the Token Dimension. As shown in Figure 1, significant energy concentration (i.e., a few adjacent frequency coefficients capture the vast majority of the energy in the key spectrum) emerges only when applying DCT along the **token dimension** of the key cache. In contrast, neither the value cache across both dimensions nor the key cache along the channel dimension exhibits this characteristic. This indicates that the token dimension is the optimal direction for key compression, corroborating the empirical finding in KIVI (Liu et al., 2024b) that per-channel quantization is preferable for keys.

Observation 2: Dynamic Range Reduction. Apart from the energy peaks, the dynamic range of the residual signal is significantly reduced compared to the original time-domain signal. This is reasonable since the total energy is conserved be-

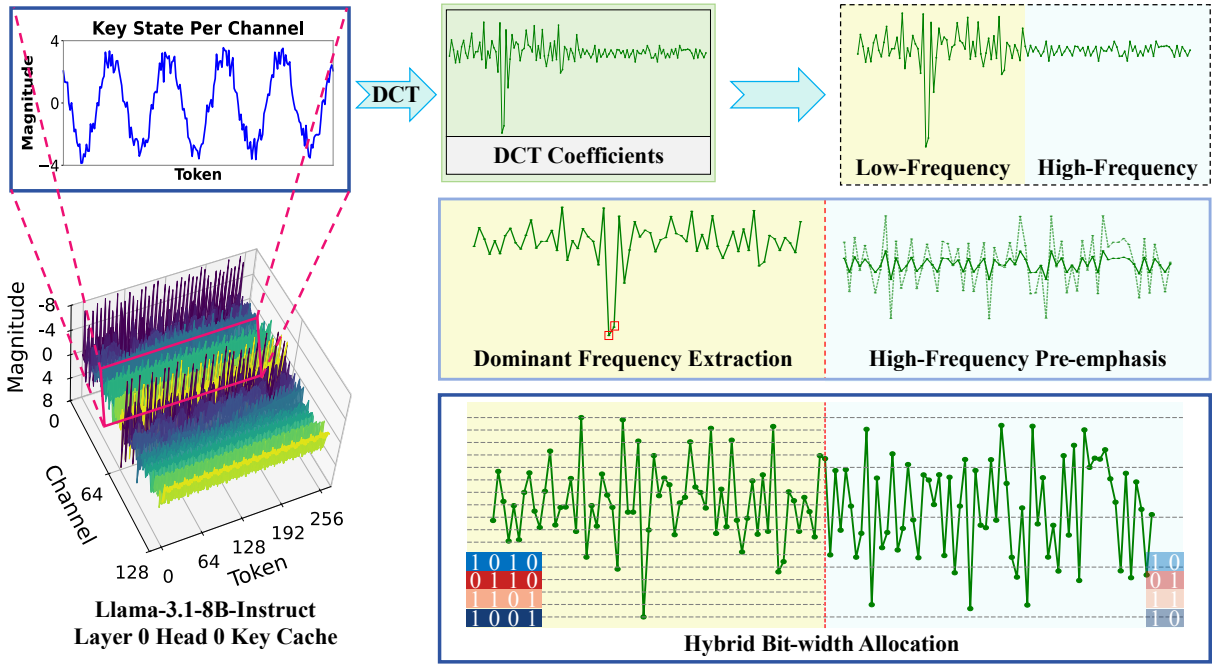

Figure 2: Overview of our SpectrumQuant framework. We transform the Key cache into the frequency domain via DCT along the token dimension. The pipeline consists of three key components: Dominant Frequency Extraction to capture concentrated energy, High-Frequency Pre-emphasis to align dynamic ranges, and Hybrid Bit-width Allocation to efficiently compress different frequency bands.

tween the time and frequency domains according to **Parseval's Theorem** (Oppenheim et al., 1997). This phenomenon directly motivates us to process the energy peaks and the residual signal respectively, i.e., quantizing the residual signal with a significantly smaller quantization scaling factor s, thereby drastically minimizing quantization errors.

Observation 3: Structural Consistency. Furthermore, we find that the dominant frequency (peak position) of a specific channel remains nearly identical across different layers and heads (see Appendix A). This consistency, determined by the RoPE frequencies (proved in Proposition 1), provides a reliable basis for designing a robust quantization strategy.

3.2 Theoretical Analysis

To provide a theoretical foundation for our approach, we model the key state as a semantic signal modulated by RoPE. Let the group size be G and the RoPE frequency for channel c be θ_c, and assuming the semantic signal (i.e., the pre-RoPE key state) varies smoothly along the token dimension, we have (detailed proofs are in Appendix J):

Proposition 1 (Spectral Shifting)**.** *RoPE modulation shifts the spectral energy center in channel c to the dominant frequency $k^* = \frac{G\theta_c}{\pi}$.*

Proposition 2 (Energy Concentration)**.** *Despite spectral leakage, for the post-RoPE key state, the two DCT coefficients nearest to k^* capture a theoretical lower bound of $\frac{6}{\pi^2}(\approx 61\%)$ of the total spectral energy.*

Proposition 3 (Stability of Spectral Coefficients)**.** *The spectral distortion is strictly bounded by the variance of the semantic signal.*

3.3 Frequency-Domain Hybrid Quantization

Based on the above observations and analyses, SpectrumQuant quantizes the key cache in the frequency domain. Let the group size be G. For a group of key cache $\mathbf{K} \in \mathbb{R}^{G\times d}$, we first apply the DCT along the token dimension to obtain the spectral coefficient matrix $\mathbf{C} = \mathrm{DCT}(\mathbf{K}) \in \mathbb{R}^{G\times d}$. The quantization process consists of the following three steps (illustrated in Figure 2):

Dominant Frequency Extraction We first isolate the highest-energy components in the spectrum. For each channel j, we extract the N coefficients with the largest magnitudes as the set of dominant frequencies $\Omega_j = \mathrm{Top}N(|\mathbf{C}_{:,j}|)$. The values $\mathbf{O}_{val}$ and indices $\mathbf{O}_{idx}$ of these coefficients are stored

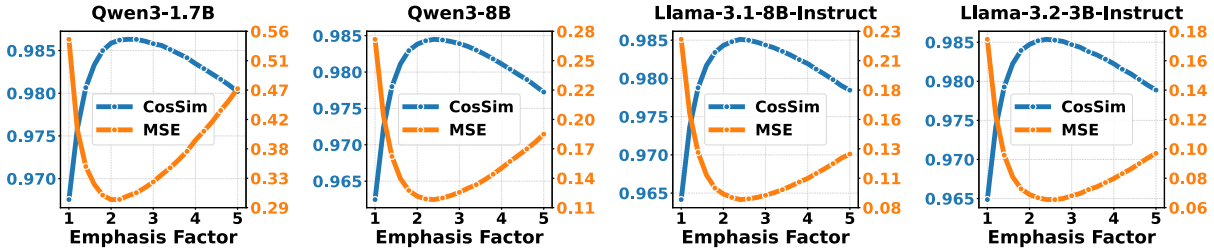

Figure 3: Impact of the emphasis factor σ on reconstruction quality of attention score across different models with the number of dominant frequencies fixed at $N = 2$ (Left Axis: Cosine Similarity (↑ higher is better), Right Axis: Mean Squared Error (↓ lower is better)).

in 16-bit and 8-bit respectively. We then zero out these positions to obtain the residual coefficients $\mathbf{R}$ with significantly reduced dynamic range:

$$\mathbf{R}_{i,j} = \begin{cases} 0, & \text{if } i \in \Omega_j \\ \mathbf{C}_{i,j}, & \text{otherwise} \end{cases} \tag{4}$$

Hybrid Bit-width Allocation At this step, we evenly split the residual spectrum $\mathbf{R}$ into a low-frequency band $\mathbf{R}_L$ (indices 1 to $G/2$) and a high-frequency band $\mathbf{R}_H$ (indices $G/2+1$ to G). We select 0.5 as the split ratio based on the theoretical result in Proposition 1 that the peak position for any channel does not exceed $\frac{G}{\pi} \approx 0.32G$, regardless of the RoPE base b. Thus, this split ensures that all dominant frequencies and their leakage energy are encapsulated within the low-frequency band. Ablation studies on split ratio are conducted in Appendix D.1.

We adopt a mixed-precision strategy: $\mathbf{R}_L$ is quantized with **4 bits**, and $\mathbf{R}_H$ with **2 bits**. This allocation stems from the insight that low-frequency distortions can corrupt the global signal structure, while errors of the same magnitude in the high-frequency band typically manifest as negligible local noise. Experiments in Appendix D.2 demonstrate that retaining 2-bit high-frequency components significantly improves reconstruction quality compared to directly discarding them.

High-Frequency Pre-emphasis and Shared Quantization To minimize metadata overhead, we let the low and high bands share a common set of quantization parameters (scaling factor s and zero-point z). However, due to the smaller dynamic range of high-frequency coefficients, direct sharing results in insufficient utilization of the quantization grid, which may lead to severe loss of high-frequency signals.

To address this issue, we introduce the **pre-emphasis techniques** from the signal processing field (López-Espejo et al., 2024), and adopt an emphasis factor σ before quantization:

$$\tilde{\mathbf{R}}_H = \sigma \cdot \mathbf{R}_H \tag{5}$$

This operation stretches the distribution of high-frequency signals to better match the dynamic range of low-frequency signals. As shown in Figure 3, reconstruction quality is optimal and robust when $\sigma \in [2.0, 3.0]$.

Subsequently, we compute the shared s and z based on the concatenated data $[\mathbf{R}_L; \tilde{\mathbf{R}}_H]$. To adapt to the difference in quantization levels between bit-widths, we introduce an adaptation factor $\gamma = \frac{2^2-1}{2^4-1} = \frac{3}{15} = \frac{1}{5}$ during mapping, which ensures that under a shared dynamic range, high-frequency data is correctly mapped to the 2-bit integer range $[0, 3]$. The final quantization formulas are:

$$\mathbf{Q}_L = \text{Clamp}\left(\left\lfloor \frac{\mathbf{R}_L - z}{s} \right\rceil, 0, 15\right) \tag{6}$$

$$\mathbf{Q}_H = \text{Clamp}\left(\left\lfloor \frac{\tilde{\mathbf{R}}_H - z}{s} \cdot \gamma \right\rceil, 0, 3\right) \tag{7}$$

And the corresponding dequantization procedure during the decoding stage is given by:

$$\hat{\mathbf{R}}_L = \mathbf{Q}_L \cdot s + z \tag{8}$$

$$\hat{\mathbf{R}}_H = \frac{(\mathbf{Q}_H / \gamma) \cdot s + z}{\sigma} \tag{9}$$

3.4 Implementation

To achieve efficient inference in long-context scenarios, we developed a set of deeply customized kernels using Triton (Tillet et al., 2019), featuring two key highlights (details in Appendix F, Algorithms 1 and 2):

Table 1: Long-context quantization evaluation on the LongBench benchmark. We report results for multiple state-of-the-art LLMs across diverse LongBench tasks.

		Single Doc. QA			Multi Doc. QA			Summarization			Few-shot Learning			
Method	Bits	NtrvQA	Qasper	MF-en	2Wiki	Hotpot	Musique	GovRep	QMSum	MNews	TREC	TriviQA	SamSum	Avg.
Qwen-3-1.7B														
fp16 Baseline	16	18.37	24.78	45.87	32.52	39.31	17.92	30.49	22.99	25.04	73.00	85.47	42.38	38.17
ZipCache	4.33	**18.65**	**25.08**	45.55	32.72	**39.25**	16.65	20.02	22.29	24.48	72.50	**86.22**	42.49	37.16
KIVI	4.33	18.58	24.99	**45.96**	32.57	38.70	**18.56**	30.55	23.21	25.00	73.00	85.56	42.01	**38.22**
PolarQuant	4.33	18.29	24.88	44.85	**32.96**	37.87	17.51	30.53	**23.32**	**25.35**	**73.50**	85.47	41.35	37.99
SpectrumQuant	3.71	18.48	24.85	45.56	32.45	38.02	17.15	**30.66**	22.90	24.59	**73.50**	85.59	**42.58**	38.02
Qwen-3-8B														
fp16 Baseline	16	26.91	48.04	53.09	43.37	58.93	36.30	33.45	23.94	24.91	72.00	89.92	42.85	46.14
ZipCache	4.33	26.92	47.30	52.23	43.45	58.86	36.37	33.18	23.59	24.78	70.00	**90.46**	44.45	45.97
KIVI	4.33	26.43	47.67	52.91	43.29	59.19	36.11	33.55	**24.48**	24.75	**71.50**	89.67	43.00	46.05
PolarQuant	4.33	26.16	47.25	**53.46**	43.42	58.91	**36.63**	**33.71**	24.21	**25.07**	**71.50**	89.75	43.22	46.11
SpectrumQuant	3.71	**27.25**	**47.80**	52.64	**43.87**	**59.58**	35.92	33.16	23.94	24.56	71.00	89.80	**44.58**	**46.18**
Llama-3.1-8B-Instruct														
fp16 Baseline	16	31.63	46.58	56.89	48.96	58.10	31.57	34.44	25.23	26.99	74.00	92.64	43.22	47.52
ZipCache	4.33	31.61	45.64	56.70	48.42	58.39	**32.07**	34.22	**25.75**	**27.03**	73.50	91.61	43.68	47.38
KIVI	4.33	32.27	**46.81**	56.74	48.82	58.08	31.83	34.49	25.07	26.73	73.50	92.27	43.84	47.53
PolarQuant	4.33	**32.50**	46.16	56.73	48.97	57.91	31.67	34.04	25.47	26.76	**74.00**	**92.86**	43.07	47.51
SpectrumQuant	3.71	32.09	46.32	**56.75**	**49.46**	**58.49**	31.58	**34.57**	25.04	26.45	**74.00**	92.21	**44.02**	**47.58**
Llama-3.2-3B-Instruct														
fp16 Baseline	16	25.60	40.85	50.67	39.74	53.19	26.08	33.35	24.68	25.66	74.50	88.78	43.04	43.85
ZipCache	4.33	25.79	**40.98**	50.00	**40.25**	52.37	**25.97**	**33.36**	**24.20**	**26.17**	**75.50**	88.78	42.13	43.79
KIVI	4.33	25.10	40.07	51.33	39.74	**53.15**	25.61	33.10	24.16	25.81	74.50	88.34	42.66	43.63
PolarQuant	4.33	25.26	40.90	50.57	39.74	52.99	25.46	32.70	23.87	26.00	74.00	88.78	42.16	43.54
SpectrumQuant	3.71	**25.92**	39.58	**51.68**	40.00	52.89	25.51	33.20	**24.20**	**26.17**	75.00	**88.88**	**42.69**	**43.81**

(1) *Operator Fusion*: In the quantization stage, minmax calculation, scaling, rounding, and bit packing are merged into a single kernel to avoid repeated global memory access of intermediate variables. In the decoding stage, we designed a fused kernel that executes dequantization, frequency-domain dot product, Inverse DCT (IDCT), and sparse accumulation within a single kernel launch.

(2) *On-Chip Computation*: We pre-load the DCT basis into the L1 cache (SRAM) and utilize broadcasting for low-latency access. For the sparse dominant frequencies, the kernel calculates and accumulates their contribution directly in registers, fully leveraging the sparsity for bandwidth efficiency.

4 Experiments

4.1 Setup

Models We conduct a systematic evaluation of SpectrumQuant across several advanced large language models, including Qwen3-1.7B, Qwen3-8B (Yang et al., 2025), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and Llama-3.2-3B-Instruct (Meta, 2024), spanning a diverse range of architectures and model sizes. Notably, the Qwen3 models introduce a pre-RoPE normalization design, distinguishing them from the Llama-3 series. All models adopt grouped query attention (GQA (Ainslie et al., 2023)).

Tasks This study primarily evaluates the performance of SpectrumQuant on long-context tasks. To this end, we adopt LongBench (Bai et al., 2023), a popular multi-task long-text benchmark, and conduct a comprehensive comparison between SpectrumQuant and mainstream high-fidelity training-free key cache quantization methods.

In addition, to verify generality, we also assess performance under standard context lengths with IF-Eval (Zhou et al., 2023), MMLU (Hendrycks et al., 2021), and GSM8K (Cobbe et al., 2021) datasets. For MMLU and GSM8K, we use 5-shot in-context learning and construct evaluation prompts with chain-of-thought (CoT). For IF-Eval, we adopt a zero-shot evaluation setting.

Implementation Details Following KIVI (Liu et al., 2024b), we perform quantization solely on the cached history after the attention computation. Consequently, the prefilling phase retains full precision and directly leverages FlashAttention-2 (Dao, 2023) for acceleration. To ensure a fair comparison, all methods retain the most recent 128 tokens in full precision, and we use a group size of 128 for all

Table 2: Performance comparisons on standard context-length benchmarks, including IF-Eval (instruction following), MMLU (knowledge and reasoning), and GSM8K (mathematical reasoning), demonstrating that our quantization method does not cause significant degradation under standard context lengths.

Model	Method	IF-Eval		MMLU					GSM8K
		Inst-Strict	Prompt-Strict	Humanities	Social Science	STEM	Other	Avg.	
Qwen-3-1.7B	fp16 Baseline	76.26	68.02	51.56	69.58	68.76	64.31	62.19	67.52
	ZipCache	73.14	64.70	48.59	65.84	59.21	59.22	57.11	39.19
	KIVI	76.62	68.39	51.03	69.13	67.75	63.82	61.58	67.07
	PolarQuant	74.94	66.91	47.86	67.18	62.04	61.60	58.32	57.26
	SpectrumQuant	76.26	68.21	50.84	69.00	66.03	64.31	61.21	66.57
Qwen-3-8B	fp16 Baseline	86.69	80.41	65.48	84.27	81.26	79.02	76.14	87.86
	ZipCache	87.17	81.33	64.72	83.33	80.21	78.47	75.31	87.14
	KIVI	87.89	82.07	65.04	83.59	80.34	78.40	75.49	86.54
	PolarQuant	86.93	80.40	64.02	82.71	80.18	78.40	74.93	86.20
	SpectrumQuant	87.53	82.07	64.31	83.56	80.24	79.08	75.37	86.97
Llama-3.1-8B-Instruct	fp16 Baseline	81.53	73.57	63.66	77.09	69.43	76.86	70.82	78.47
	ZipCache	80.46	72.83	63.06	78.39	69.97	77.08	71.07	79.94
	KIVI	81.77	74.12	64.08	77.64	70.66	75.89	71.14	79.38
	PolarQuant	79.85	72.82	63.85	77.19	70.22	76.92	71.09	79.61
	SpectrumQuant	82.25	75.05	63.38	76.96	68.51	75.99	70.30	79.91
Llama-3.2-3B-Instruct	fp16 Baseline	79.62	72.09	54.86	66.75	63.40	67.91	62.27	70.36
	ZipCache	78.54	69.69	53.45	66.92	61.05	67.65	61.25	70.12
	KIVI	77.82	69.87	53.11	67.83	61.50	67.81	61.47	69.98
	PolarQuant	79.62	71.53	53.01	66.10	60.48	68.36	60.95	69.83
	SpectrumQuant	78.66	71.06	53.94	67.34	61.75	68.20	61.79	69.90

Table 3: End-to-end inference latency (seconds) and peak GPU memory usage (GB) on LongBench subtasks, evaluated with Llama-3.1-8B-Instruct on two NVIDIA RTX 5090 GPUs.

Method	Latency (s) / Mem. (GB)						
	NtrvQA	Qasper	MF-en	Musique	GovRep	QMSum	MNews
FP16 Baseline	877.5 / 54.07	211.8 / 22.81	171.4 / 20.50	351.7 / 21.02	2880.1 / 40.81	1018.2 / 32.13	2630.1 / 20.12
ZipCache	**827.1** / 37.58	215.5 / 20.26	177.6 / 18.70	347.0 / 19.05	**2745.3** / 27.75	1108.5 / 22.53	2811.6 / 18.44
KIVI	842.0 / 37.68	222.7 / 20.38	179.0 / 18.79	**346.3** / 19.10	2793.5 / 27.76	1058.3 / 22.71	2717.1 / 18.55
PolarQuant	891.6 / 38.23	244.8 / 20.46	197.2 / 18.84	374.8 / 19.20	4055.7 / 28.23	1297.0 / 22.82	3257.6 / 18.57
SpectrumQuant (PyTorch)	904.4 / 37.44	269.1 / 20.22	209.8 / 18.67	389.9 / 19.02	4256.9 / 27.67	1442.2 / 22.49	3390.3 / 18.42
SpectrumQuant (Triton)	829.0 / **37.44**	**209.2 / 20.22**	**173.8 / 18.67**	353.9 / **19.02**	2771.1 / **27.67**	**1057.8 / 22.49**	**2669.7 / 18.42**

group-wise methods. Following PolarQuant (Wu et al., 2025), we evaluate all comparison methods using 4-bit quantization. For SpectrumQuant, we set the number of dominant frequencies $N = 2$ (consistent with Proposition 2) and the emphasis factor $\sigma = 2.0$ based on our ablation studies (Section 4.4). Regarding the evaluation framework, we align our LongBench evaluation setting with KIVI, while employing lm-evaluation-harness (Gao et al., 2024a) for other benchmarks. All experiments were conducted on two NVIDIA GeForce RTX 5090 GPUs.

4.2 Main Results

Table 1 summarizes the performance on LongBench. SpectrumQuant demonstrates strong capabilities across diverse tasks, yielding results comparable to existing high-fidelity approaches while handling long-context inputs.

We further evaluate SpectrumQuant on standard context benchmarks, as summarized in Table 2. SpectrumQuant maintains competitive performance across knowledge-intensive (MMLU), mathematical reasoning (GSM8K), and instruction-following (IF-Eval) tasks without notable degradation. This confirms that our quantization scheme preserves the model's fundamental capabilities across general-purpose tasks. Notably, SpectrumQuant exhibits **superior robustness** across diverse model architectures: it restricts the maximum accuracy drop to merely **1.6%** across all tasks on all tested models, confirming its reliability as a general quantization scheme.

4.3 Efficiency

(1) *Memory Usage.* As shown in Table 3, SpectrumQuant effectively alleviates memory bottlenecks by substantially reducing peak GPU memory

Table 4: LongBench results of SpectrumQuant combined with token-wise value cache quantization.

Method	Value Bits	Single Doc. QA			Multi Doc. QA			Summarization			Few-shot Learning			Avg.
		NtrvQA	Qasper	MF-en	2Wiki	Hotpot	Musique	GovRep	QMSum	MNews	TREC	TriviQA	SamSum	
SpectrumQuant	16	32.09	46.32	56.75	49.46	58.49	31.58	34.57	25.04	26.45	74.00	92.21	44.02	47.58
	4	32.10	45.76	56.15	49.30	58.73	31.69	34.51	25.48	26.89	74.00	91.83	43.86	47.52
	2	32.53	45.15	56.70	47.31	58.42	31.58	34.07	25.47	26.78	73.50	91.72	44.26	47.29

Table 5: Ablation studies on the number of dominant frequencies N and the emphasis factor σ. The reported values are the average scores on LongBench subtasks for Llama-3.2-3B-Instruct, Qwen3-8B, and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023).

Method	N	σ	Avg Scores		
			Llama-3.2-3B-Instruct	Qwen3-8B	Mistral-7B-Instruct-v0.3
FP16 Baseline	None	None	37.25	41.43	37.60
w/o DCT	2	1.0	36.51	40.81	36.16
SpectrumQuant	1	1.0	18.62	29.81	21.24
	1	2.0	36.68	41.00	37.28
	1	3.0	36.93	41.04	37.24
	2	1.0	36.34	39.89	36.16
	2	2.0	**37.23**	<u>41.37</u>	**37.52**
	2	3.0	37.14	41.20	<u>37.48</u>
	3	1.0	36.24	40.90	37.12
	3	2.0	<u>37.21</u>	**41.41**	37.39
	3	3.0	37.17	41.29	37.31

usage, with the advantage particularly evident in long-context scenarios. For a fair comparison of peak memory usage across all methods, we disable the fused quantization-packing kernels in both KIVI and our Triton implementation. Enabling these kernels would further reduce the peak memory footprint by 5% to 8%. We provide a detailed calculation of the actual average bitwidth (including all metadata and the full-precision residual length) for all evaluated methods in Appendix C.

(2) *Inference Latency.* We compare the end-to-end latency of a baseline using a 16-bit floating-point KV cache with FlashAttention-2 (Dao, 2023) against SpectrumQuant implemented in both PyTorch and Triton, as well as other state-of-the-art quantization methods (see Table 3). Notably, the overhead of the naive PyTorch implementation becomes increasingly pronounced on tasks with ultra-long contexts, where the latency gap widens significantly. In contrast, our Triton kernels successfully eliminate this bottleneck, achieving low latency while significantly reducing peak GPU memory usage across all tasks.

4.4 Analysis

Performance under variant RoPE configurations In Appendix D.3, we further evaluate SpectrumQuant on Llama-2-7B-32K (Chen et al., 2023). Collectively, our experiments span diverse RoPE configurations, including Llama-2-7B-32K (base $b = 10000$, Linear Scaling), Llama-3.1 model ($b = 500000$, YaRN-like (Peng et al., 2024) Scaling), and Qwen3 series ($b = 1000000$, standard RoPE). The consistent performance across these settings proves the robustness of SpectrumQuant to varying base frequencies and RoPE scaling strategies.

Compatibility with value cache quantization Following PolarQuant (Wu et al., 2025), we investigate integrating SpectrumQuant with value cache quantization to further minimize memory footprint. We employ standard token-wise quantization for values, consistent with KIVI (Liu et al., 2024b). As presented in Table 4, even under aggressive 2-bit value quantization, the performance degradation remains within an acceptable range. This confirms that our method is orthogonal to and compatible with existing value quantization techniques.

Table 6: Experimental results for SpectrumQuant on LongBench tested with Qwen3-8B-AWQ model.

		Single Doc. QA			Multi Doc. QA			Summarization		
Method	Bits	NtrvQA	Qasper	MF-en	2Wiki	Hotpot	Musique	GovRep	QMSum	Avg.
fp16 Baseline	16	26.86	45.72	50.91	41.49	58.60	30.27	33.59	23.61	38.88
SpectrumQuant	3.71	26.14	45.06	49.90	42.13	59.22	29.90	33.54	23.52	38.68

Combination with Weight Quantization Our framework is orthogonal to weight quantization and can be seamlessly integrated with techniques such as AWQ (Lin et al., 2024). Empirically, we observe that the energy concentration phenomenon in the DCT domain persists even when the model parameters are quantized. The experimental results on LongBench subtasks for the Qwen3-8B-AWQ model are presented in Table 6. Overall, integrating SpectrumQuant with AWQ further shrinks the memory footprint during inference. Crucially, it preserves the performance of the weight-quantized backbone with negligible degradation. This validates the robustness and excellent compatibility of our quantization approach across different precisions of model weights.

Ablation studies We conduct ablation studies on the core components of SpectrumQuant (see Table 5). The results show that, under the same configuration, SpectrumQuant significantly outperforms a time-domain baseline without DCT, where we set $\sigma = 1.0$ as frequency-dependent pre-emphasis is inapplicable in the time domain, confirming the critical role of the frequency-domain transform in compression effectiveness.

Experiments on the number of dominant frequencies N indicate that retaining only the strongest coefficient ($N = 1$) causes substantial performance degradation; in contrast, $N = 2$ is enough to recover most of the accuracy, while increasing to $N = 3$ yields only marginal gains. Therefore, we choose $N = 2$ to balance performance and overhead. Moreover, introducing the emphasis factor σ effectively improves quantization performance. The model maintains stable accuracy for $\sigma \in [2.0, 3.0]$, consistent with our reconstruction experiments (see Fig. 3), validating the necessity of using the emphasis technique to compensate for high-frequency quantization errors.

In Appendix D, we further investigate the impact of group size (§D.4) and evaluate SpectrumQuant on the Needle-in-a-Haystack benchmark for ultra-long context scenarios (§D.5).

5 Conclusion

In this paper, we introduced SpectrumQuant, a frequency-domain framework that resolves the challenge of quantizing RoPE-modulated key caches. By leveraging the energy concentration properties of DCT, SpectrumQuant achieves efficient compression through dominant frequency extraction and hybrid bit-width allocation. Furthermore, our hardware-optimized fused kernels eliminate the computational overhead. Extensive experiments verify that SpectrumQuant greatly reduces memory usage while maintaining performance comparable to FP16 baselines, offering an efficient solution for long-context LLM inference.

Limitations

Despite the effectiveness of SpectrumQuant, several avenues remain for future exploration. First, our framework currently relies exclusively on the discrete cosine transform; future work could investigate other spectral transformation tools (e.g., wavelets) to further optimize energy concentration. Second, while our fused Triton kernels ensure efficiency, we believe there is room to design even stronger kernels to further minimize inference latency. Finally, we have not yet applied this frequency-domain perspective to model weight quantization, which remains a promising direction for developing a unified spectral compression framework.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–

4901, Singapore. Association for Computational Linguistics.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Ningning Chen, Weicai Ye, and Ying Jiang. 2025. Hbllm: Wavelet-enhanced high-fidelity 1-bit quantization for llms. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024a. The language model evaluation harness.

Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. 2024b. Parameter-efficient fine-tuning with discrete fourier transform. In *International Conference on Machine Learning*, pages 14884–14901. PMLR.

gkamradt. 2023. Llmtest needle in a haystack - pressure testing llms. `https://github.com/gkamradt/LLMTest_NeedleInAHaystack`.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yefei He, Luoming Zhang, Weijia Wu, Jing Liu, Hong Zhou, and Bohan Zhuang. 2024. Zipcache: Accurate and efficient kv cache quantization with salient token identification. *Preprint*, arXiv:2405.14256.

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. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8954–8966, Toronto, Canada. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

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. *Preprint*, arXiv:2401.18079.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Jushi Kai, Boyi Zeng, Yixuan Wang, Haoli Bai, Ziwei He, Bo Jiang, and Zhouhan Lin. 2025. Freqkv: Frequency domain key-value compression for efficient context window extension. *Preprint*, arXiv:2505.00570.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontañón. 2022. FNet: Mixing tokens with Fourier transforms. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4296–4313, Seattle, United States. Association for Computational Linguistics.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6:87–100.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. 2024a. Spinquant: Llm quantization with learned rotations. In *The Thirteenth International Conference on Learning Representations*.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024b. Kivi: a tuning-free asymmetric 2bit quantization for kv cache. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, pages 32332–32344.

Iván López-Espejo, Aditya Joglekar, Antonio M. Peinado, and Jesper Jensen. 2024. On speech pre-emphasis as a simple and inexpensive method to boost speech enhancement. *Preprint*, arXiv:2401.09315.

Meta. 2024. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models.

Alan V Oppenheim, Alan S Willsky, and Syed Hamid Nawab. 1997. *Signals & systems*. Pearson Educación.

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*.

Jorn Peters, Marios Fournarakis, Markus Nagel, Mart Van Baalen, and Tijmen Blankevoort. 2023. Qbitopt: Fast and accurate bitwidth reallocation during training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1282–1291.

Valeria Ruscio, Umberto Nanni, and Fabrizio Silvestri. 2025. Beyond position: the emergence of wavelet-like properties in transformers. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6074–6088.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding. *Preprint*, arXiv:2104.09864.

Philippe Tillet, H. T. Kung, and David Cox. 2019. Triton: an intermediate language and compiler for tiled neural network computations.

Songhao Wu, Ang Lv, Xun Zhang, Guojun Yin, Wei Lin, and Rui Yan. 2025. Polarquant: Leveraging polar transformation for key cache quantization and decoding acceleration. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

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*.

Zhixiong Zhao, Fangxin Liu, Junjie Wang, Chenyang Guan, Zongwu Wang, Li Jiang, and Haibing Guan. 2026. Specquant: Spectral decomposition and adaptive truncation for ultra-low-bit llms quantization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 28786–28794.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

Appendix

A Visualization of Structural Consistency

Figure 4 illustrates the DCT spectrum of key states sampled from various layers and attention heads. The visualization reveals a clear pattern of structural consistency: within the same channel index (column), the dominant frequency peaks emerge at identical positions across different layers and heads (rows). This phenomenon confirms that the spectral sparsity pattern is intrinsically governed by the RoPE rotation frequencies, supporting the rationale for applying a unified quantization strategy globally.

B Discrete Cosine Transform

The Discrete Cosine Transform (DCT) is an orthogonal transform. It is widely used in signal processing and image compression standards, such as JPEG. In this paper, we adopt the most common variant, DCT-II.

For a real-valued sequence of length L, denoted as $\mathbf{X} = [x_0, x_1, \ldots, x_{L-1}] \in \mathbb{R}^L$, its forward DCT-II transform is defined as the coefficient vector $\mathbf{C} \in \mathbb{R}^L$:

$$c_k = \alpha_k \sum_{n=0}^{L-1} x_n \cdot \cos\left[\frac{\pi}{L}\left(n+\frac{1}{2}\right)k\right], \quad k = 0, \ldots, L-1 \tag{10}$$

where α_k is the normalization coefficient ensuring the orthogonality of the transform:

$$\alpha_k = \begin{cases} \sqrt{\frac{1}{L}}, & \text{if } k = 0 \\ \sqrt{\frac{2}{L}}, & \text{otherwise} \end{cases} \tag{11}$$

Since the DCT is an orthogonal transform, its inverse (IDCT) is simply its transpose. The original signal $\mathbf{X}$ can be losslessly reconstructed via:

$$x_n = \sum_{k=0}^{L-1} \alpha_k \cdot c_k \cdot \cos\left[\frac{\pi}{L}\left(n+\frac{1}{2}\right)k\right], \quad n = 0, \ldots, L-1 \tag{12}$$

The core advantage of DCT lies in its **energy concentration** property. Since the rotational modulation of RoPE is structurally homologous to the cosine basis functions of DCT, DCT can effectively deconstruct the intense time-domain oscillations of key cache into sparse impulses in the frequency domain. This property allows us to precisely isolate high-energy dominant frequencies. Consequently, we can significantly reduce the quantization scaling factor of the remaining coefficients, laying the foundation for our quantization method.

C Details of Average Bit-Width Calculation

To fairly evaluate the memory efficiency of SpectrumQuant, we follow the estimation procedure of PolarQuant (Wu et al., 2025). We set the input sequence length $T = 12.2K$ (the average input length on LongBench), the attention head dimension $d = 128$, the quantization group size $G = 128$, and residual length (length of the retained full-precision buffer) $s = 128$. To simplify the formulation, we omit the batch and head dimensions.

Under this setting, our storage overhead consists of the following four components:

- **Spectral Data**: The low-frequency band ($G/2$ coefficients) employs 4-bit quantization, and the high-frequency band ($G/2$ coefficients) uses 2-bit quantization. This results in an average data bit-width of 3.0 bits/token.
- **Dominant Frequency Overhead**: For N dominant frequency components extracted per group, we store N full-precision values and N 8-bit indices, totaling $24N$ bits. As we set $N = 2$ in our experiments, the amortized cost is $\frac{24N}{G} = 0.375$ bits/token.
- **Group Metadata**: Each group shares a single set of full-precision Scaling Factor (Scale) and Zero-point (Min). The overhead is $\frac{32}{G} = 0.25$ bits/token.
- **Full-Precision Buffer**: Following KIVI (Liu et al., 2024b), we retain a full-precision buffer of length s to maintain streaming generation quality. In long-context scenarios ($T \gg s$), the expected occupied length is $s/2$. This introduces a marginal overhead of approximately $\frac{16\times(s/2)}{T} \approx 0.08$ bits/token.

In summary, the total effective bit-width of SpectrumQuant is $3.0+0.375+0.25+0.08 \approx \mathbf{3.71}$ bits. This result demonstrates that SpectrumQuant offers a storage-efficient solution while maintaining robust model performance.

For comparison, we also calculate the actual bit-width of other state-of-the-art quantization methods. For quantization parameters, ZipCache (He

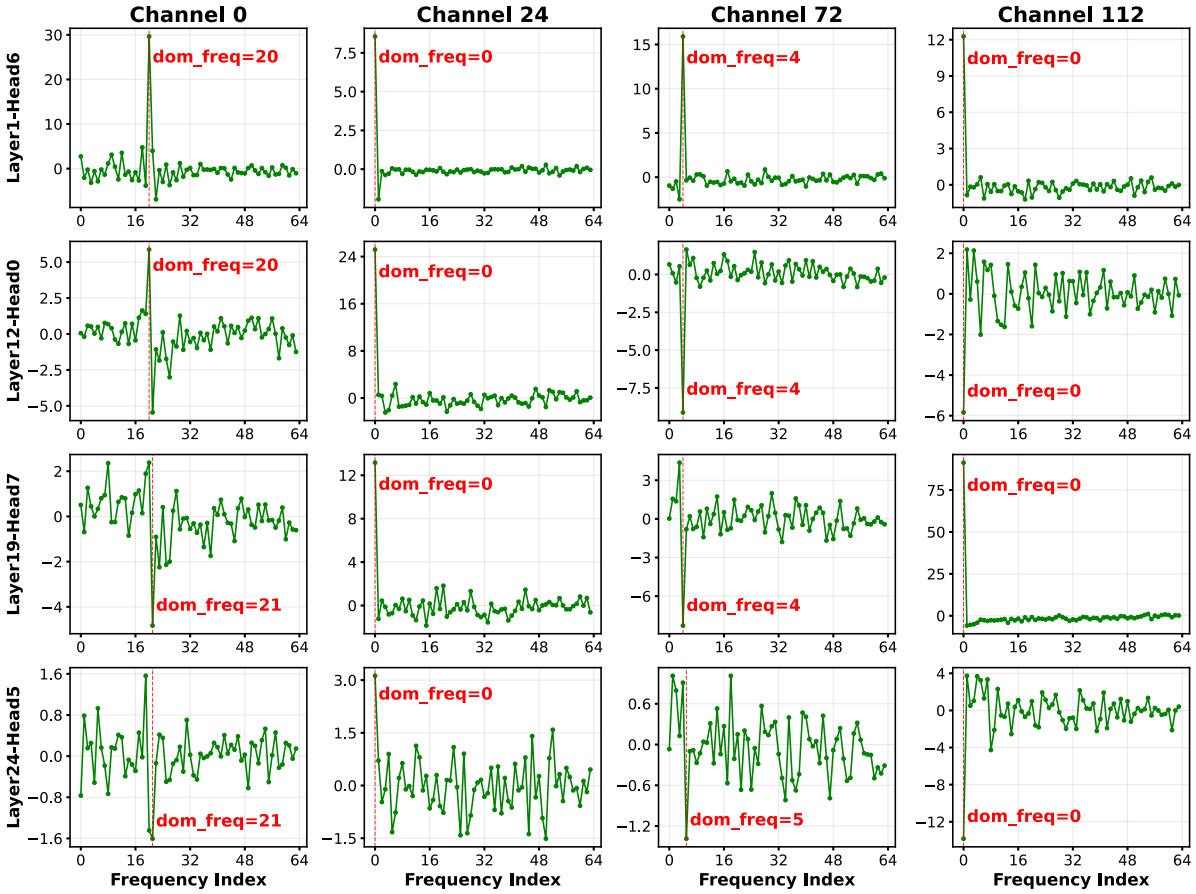

Figure 4: Visualization of Structural Consistency across layers and heads.

Table 7: LongBench results on Llama-2-7B-32K, demonstrating the robustness of SpectrumQuant under Position Interpolation.

		Single Doc. QA			Multi Doc. QA			Summarization		
Method	Bits	NtrvQA	Qasper	MF-en	2Wiki	Hotpot	Musique	GovRep	QMSum	Avg.
fp16 Baseline	16	14.71	12.31	20.76	11.80	13.89	7.29	30.51	21.69	16.62
SpectrumQuant	3.71	15.58	11.73	20.53	11.32	14.04	7.67	29.28	21.80	16.49

et al., 2024) introduces a channel-separable, token-wise scheme, incurring $32/d$ bits (16 bits for zero-points and 16 bits for scaling factors per token), which amounts to 0.25 bits/token when $d = 128$. Conversely, KIVI (Liu et al., 2024b) and PolarQuant (Wu et al., 2025) utilize channel-wise quantization, introducing $32d$ bits of parameters per group, which increases the average bit-width by $32/G = 0.25$ bits/token. As calculated above, all three methods require the same full-precision buffer for streaming generation, adding an identical overhead of approximately 0.08 bits/token. Considering the base cost of 4-bit quantization, the total effective bit-width for them is identically $4.0 + 0.25 + 0.08 = \mathbf{4.33}$ bits.

D Further Analysis

Consistent with configurations in main experiments, we set the number of dominant frequencies $N = 2$ and the emphasis factor $\sigma = 2.0$ through this section.

D.1 Ablation Study on Split Ratio

We evaluate the impact of different split ratios using Llama-3.1-8B-Instruct on LongBench subtasks. As shown in Table 8, setting the ratio to 0.25 leads to a noticeable performance drop. This validates our theoretical analysis: A 0.25 split fails to encapsulate all dominant frequencies and their leakage energy within the low-frequency band. Conversely, increasing the ratio to 0.75 yields marginal gains

but introduces unnecessary overhead. Therefore, we select 0.5 as the optimal balance between accuracy and memory efficiency.

Table 8: Ablation study of split ratio on LongBench subtasks with group size $G = 128$ and residual length $s = 128$.

Split Ratio	0.25	0.5	0.75
Avg. (Bits)	41.43 (3.21)	41.79 (3.71)	41.82 (4.21)

D.2 Necessity of Retaining High-Frequency Components

FreqKV (Kai et al., 2025) compresses the KV cache by discarding high-frequency DCT components and applying a rescaling factor of $\sqrt{L/N}$ (where L and N are the retained length and the original length respectively) during the inverse DCT. To analyze the efficacy of this approach, we compare our method against a truncation approach designed in the spirit of FreqKV under an iso-bitwidth setting. Specifically, the approach discards the highest 25% of frequencies and quantizes the remaining coefficients to 4 bits, matching the average bit-width of SpectrumQuant.

As shown in Figure 5, SpectrumQuant consistently outperforms the truncation scheme across all models. This demonstrates that retaining the full spectrum with low precision for high frequencies yields superior fidelity compared to truncation, confirming that these components carry non-negligible information essential for reconstruction.

D.3 Linear RoPE Scaling Experiment

To assess the robustness of SpectrumQuant under *linear RoPE scaling*, we evaluate its performance on Llama-2-7B-32K (Chen et al., 2023). This model extends the context window to 32K via Position Interpolation (PI), which linearly scales the RoPE frequencies to adapt to longer sequences.

As shown in Table 7, SpectrumQuant achieves an average score of 16.49, closely matching the FP16 baseline of 16.62 on LongBench subtasks. This demonstrates that our strategy effectively preserves information even when the spectral characteristics of RoPE are altered by linear scaling, further validating the generalizability of our approach.

D.4 Impact of Group Size G

We investigate the influence of the quantization group size G using Llama-3.1-8B-Instruct on LongBench. As shown in Table 9, smaller group sizes significantly increase metadata overhead, resulting in higher effective bit-widths without yielding performance gains. Conversely, extending the group size to 256 further reduces memory usage but leads to a slight degradation in accuracy. Consequently, we select $G = 128$ as the optimal configuration to achieve the best trade-off between reconstruction fidelity and compression efficiency, aligning with the findings in PolarQuant (Wu et al., 2025).

Table 9: Ablation study of group size on LongBench subtasks with fixed residual length $s = 256$.

Group Size	32	64	128	256
Avg. (Bits)	41.69 (5.66)	41.74 (4.41)	41.78 (3.79)	41.64 (3.47)

Table 10: Performance on the Needle-in-a-Haystack benchmark, evaluated using Llama-3.1-8B-Instruct at a **128K** context length on a single NVIDIA A800 GPU. TTPS stands for Test Time per Sample (s).

Method	Score	TTPS
FP16 Baseline	99.4	37.8
ZipCache	97.8	38.7
KIVI	98.4	**38.5**
PolarQuant	96.6	43.5
SpectrumQuant (Ours)	**98.8**	38.8

D.5 Ultra-Long Context Evaluation on Needle-in-a-Haystack

To comprehensively evaluate the capability of our method under ultra-long context scenarios, we extended our experiments to the Needle-in-a-Haystack Benchmark (gkamradt, 2023), which is designed to evaluate the long-context retrieval and memory capabilities of LLMs. Specifically, we used Llama-3.1-8B-Instruct with a **128K** context length.

As shown in Table 10, our method achieves a score highly comparable to the FP16 baseline and outperforms other quantization baselines. Notably, our method significantly reduces peak memory usage while maintaining competitive inference latency, demonstrating its efficiency in processing ultra-long contexts.

E Detailed Comparison with Relevant Frequency-Domain Methods

In this section, we provide a detailed comparison between SpectrumQuant and the most relevant re-

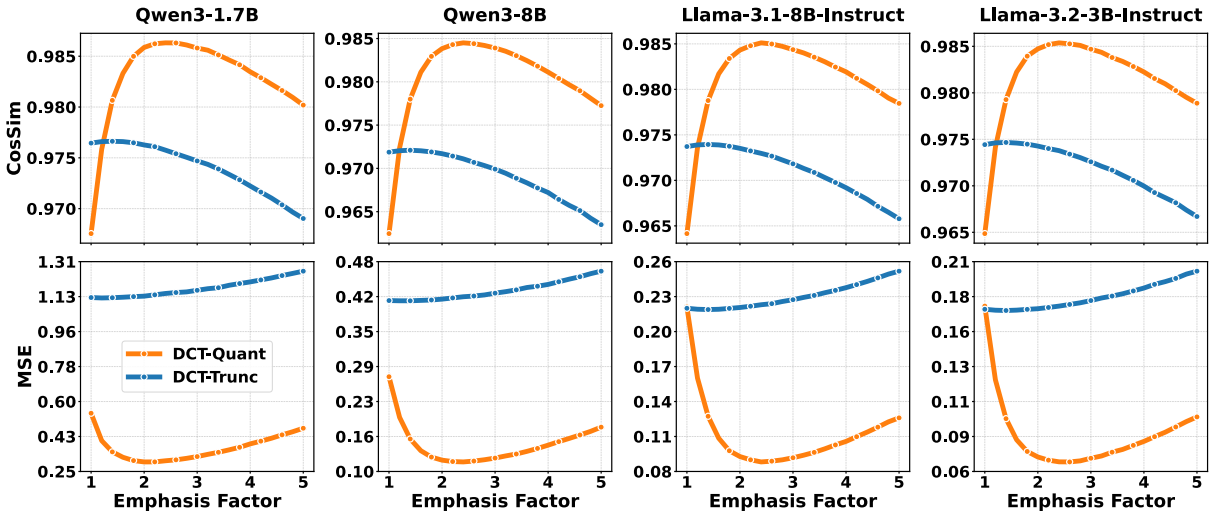

Figure 5: Reconstruction quality comparison between High-Frequency Quantization (Ours) and Truncation across different models. The top row shows Cosine Similarity (↑), and the bottom row shows Mean Squared Error (↓).

cent works, FreqKV (Kai et al., 2025) and SpecQuant (Zhao et al., 2026), to clarify our fundamental differences in scope, motivation, and methodology.

While FreqKV operates in the frequency domain, its primary objective is *token eviction* (reducing sequence length) rather than *quantization*. It discards high-frequency components, a strategy that destroys information and requires fine-tuning to recover performance. Furthermore, FreqKV avoids RoPE-induced distortions by applying generic filters only to Pre-RoPE keys. In contrast, our work tackles the challenging *Post-RoPE quantization* problem. Recognizing that discarding high-frequency components degrades reconstruction fidelity (Appendix D.2), we theoretically characterize RoPE-induced spectral shifts and propose a training-free quantization scheme that preserves all tokens.

SpecQuant utilizes spectral transformations to facilitate *weight and activation quantization* by exploiting intrinsic weight smoothness. Since its transformation is applied to weights offline, it cannot address the dynamic effects of RoPE. Methodologically, SpecQuant relies on a low-frequency truncation strategy. In contrast, our work leverages the insight that RoPE induces *temporal periodicity*, resulting in *spectral sparsity*. Rather than discarding high frequencies, we dynamically compress both low- and high-frequency bands. Furthermore, unlike SpecQuant which requires offline calibration datasets, our method is entirely calibration-free and plug-and-play.

F Details of Algorithms

The specific implementation details of SpectrumQuant are presented in Algorithm 1 and Algorithm 2. In this section, we primarily introduce two designs in the decoding kernel aimed at minimizing memory access overhead.

Deferred Inverse Transform During decoding, directly performing the Inverse Discrete Cosine Transform (IDCT) on the frequency coefficient matrix $\mathbf{C}$ to recover time-domain keys $\tilde{\mathbf{K}} = \mathcal{B}^\top \mathbf{C}$ would require explicit reconstruction of the time-domain tensor. This hinders operator fusion. To address this, we leverage the property of matrix transposition, where $\mathbf{q}$ denotes the query vector:

$$\text{Score} = \mathbf{q} \cdot \tilde{\mathbf{K}}^\top = \mathbf{q} \cdot (\mathcal{B}^\top \mathbf{C})^\top = (\mathbf{q} \cdot \mathbf{C}^\top)\mathcal{B} \quad (13)$$

In our kernel (Part 1 of Algorithm 2), we reorder the computation. We first calculate the dot product between the query vector and the dequantized coefficients in the frequency domain to obtain frequency-domain scores. Subsequently, we apply a global projection using the DCT basis $\mathcal{B}$ to transform the intermediate scores back to the time domain. We term this technique **Deferred IDCT**. This design circumvents the materialization of the fully expanded time-domain key tensor. Crucially, it mitigates the memory bandwidth bottleneck associated with the large key cache, allowing the entire decoding process to be executed efficiently.

Sparse Dominant Frequency Accumulation Since Triton lacks a native `scatter_add` primitive, relying on `atomic_add` for sparse reconstruction

would incur significant serialization overhead due to memory contention. Consequently, we fully exploit the sparsity of the extracted dominant frequencies (Part 2 of Algorithm 2). For each dominant frequency, we compute its specific contribution to the attention score (i.e., the product of the coefficient value, the corresponding query element and the basis vector at the specific frequency) and accumulate it directly into the final result. This "compute-on-the-fly" strategy completely eliminates the need for memory read/write operations for intermediate results.

G Compute Resource Details

All experiments were conducted on two NVIDIA GeForce RTX 5090 GPUs. Each GPU is equipped with 32 GB of memory. On average, each method takes roughly **4 hours** to finish a complete benchmark on a single model.

H Statistics of Datasets

We evaluate SpectrumQuant on four benchmarks using their official splits. The detailed statistics and settings are as follows:

(1) *Longbench* (Bai et al., 2023): We utilize the English subset (official test split) covering Single-Doc QA, Multi-Doc QA, Summarization, and Few-shot Learning. Most subtasks contain 200 examples (totaling ∼2,400 samples). Input lengths are truncated to fit the model's context window.

(2) *MMLU* (Hendrycks et al., 2021): Covering 57 subjects across STEM and humanities, we evaluate on the full test split (14,042 questions). We employ a 5-shot Chain-of-Thought (CoT) setting with exemplars drawn from the development split. Notably, we adopt CoT to elicit longer generation since standard MMLU answers are typically very short; this ensures the decoding process of quantization method is sufficiently triggered, facilitating a fair comparison.

(3) *GSM8K* (Cobbe et al., 2021): For mathematical reasoning, we evaluate on the test split (1,319 examples) using a 5-shot Chain-of-Thought (CoT) setting. Exemplars are randomly sampled from the training set (7,473 examples).

(4) *IF-Eval* (Zhou et al., 2023): To assess instruction following, we evaluate on the entire dataset of 541 prompts containing verifiable constraints. The evaluation is conducted in a zero-shot setting.

I LLM Usage

The authors of this paper used Google Gemini (`https://gemini.google.com/`) for polishing text within this paper. The authors take full responsibility for the content within this paper.

Algorithm 1 Fused Quantization and Packing

Require: Key cache $\mathbf{K} \in \mathbb{R}^{G \times D}$, DCT Basis $\mathcal{B} \in \mathbb{R}^{G \times G}$, group size G , Emphasis Factor σ, the number of dominant frequencies N.

▷ We omit batch and head dimensions for clarity

Ensure: Packed Keys $\mathbf{K}_L, \mathbf{K}_H$, Quantization Params $\mathbf{S}, \mathbf{Z}$, Dominant Frequency Components $\mathbf{O}_{idx}, \mathbf{O}_{val}$.

Step 1: Frequency Domain Transformation

$\mathbf{C} \leftarrow \mathcal{B} \cdot \mathbf{K}$

Step 2: Dominant Frequency Extraction (Per-Channel)

parallel For $d = 0$ to $D - 1$

  Find indices $I \in \mathbb{Z}^N$ of the N largest values in $|\mathbf{C}_{:,d}|$.

  $\mathbf{O}_{idx}[:, d] \leftarrow I, \quad \mathbf{O}_{val}[:, d] \leftarrow \mathbf{C}_{I,d}$.

  $\mathbf{C}_{I,d} \leftarrow 0$ ▷ Remove dominant frequencies from coefficients

end parallel For

Step 3: Fused Quantization Kernel

parallel For $d = 0$ to $D - 1$

  Load coefficients $\mathbf{c} \leftarrow \mathbf{C}_{:,d}$.

  Calculate min/max of $\mathbf{c}$ to get $S[d]$ and $Z[d]$.

  Split $\mathbf{c}$ into Low-freq $\mathbf{c}_L$ and High-freq $\mathbf{c}_H$.

  $\hat{\mathbf{c}}_L \leftarrow \text{Quant}(\mathbf{c}_L, S[d], Z[d], \text{4-bit})$.

  $\hat{\mathbf{c}}_H \leftarrow \text{Quant}(\mathbf{c}_H \cdot \sigma, S[d], Z[d], \text{2-bit})$ ▷ Scale high-freq before quantization

  $\mathbf{K}_L[d] \leftarrow \text{BitPack}(\hat{\mathbf{c}}_L), \quad \mathbf{K}_H[d] \leftarrow \text{BitPack}(\hat{\mathbf{c}}_H)$

end parallel For

return $\mathbf{K}_L, \mathbf{K}_H, \mathbf{S}, \mathbf{Z}, \mathbf{O}_{idx}, \mathbf{O}_{val}$

Algorithm 2 Fused Decoding Kernel

Require: Query $\mathbf{q} \in \mathbb{R}^{1 \times D}$, Packed Keys $\mathbf{K}_L, \mathbf{K}_H$, Scales $\mathbf{S}$, Zeros $\mathbf{Z}$, DCT Basis $\mathcal{B}$, Dominant Frequency Components $\mathbf{O}_{idx}, \mathbf{O}_{val}$, Emphasis Factor σ.

Ensure: Attention Score in Time Domain $\mathbf{A}_{time} \in \mathbb{R}^G$.

Shared Memory: Load Basis Matrix $\mathcal{B}$ into SRAM (L1 Cache).

Register: Initialize accumulator $\mathbf{Acc} \in \mathbb{R}^G$ to zeros. Load $\mathbf{q}, \mathbf{S}, \mathbf{Z}$.

Part 1: Vectorized Dense Computation

  Load packed blocks $\mathbf{K}_L$ and $\mathbf{K}_H$.

  $\mathbf{V}_L \leftarrow \text{VecUnpack}(\mathbf{K}_L, \text{4-bit}), \quad \mathbf{V}_H \leftarrow \text{VecUnpack}(\mathbf{K}_H, \text{2-bit})$.

  $\hat{\mathbf{F}}_L \leftarrow \mathbf{V}_L \odot \mathbf{S} + \mathbf{Z}$

  $\hat{\mathbf{F}}_H \leftarrow (5.0 \cdot \mathbf{V}_H \odot \mathbf{S} + \mathbf{Z})/\sigma$ ▷ High-freq inverse scaling with σ

  $\mathbf{s}_L \leftarrow \mathbf{q} \cdot \hat{\mathbf{F}}_L, \quad \mathbf{s}_H \leftarrow \mathbf{q} \cdot \hat{\mathbf{F}}_H$ ▷ Dot product over channel dim

  Let $\mathcal{B}_L, \mathcal{B}_H$ be the rows of $\mathcal{B}$ corresponding to low/high frequencies.

  $\mathbf{Acc} \leftarrow \mathbf{Acc} + \mathbf{s}_L \mathcal{B}_L + \mathbf{s}_H \mathcal{B}_H$ ▷ Deferred IDCT

Part 2: Sparse Component Accumulation

parallel For $d = 0$ to $D - 1$

  Load sparse index $idx \leftarrow \mathbf{O}_{idx}[:, d]$ and value $val \leftarrow \mathbf{O}_{val}[:, d]$.

  $w \leftarrow val \cdot \mathbf{q}[d]$

  $\mathbf{Acc} \leftarrow \mathbf{Acc} + w \cdot \mathcal{B}[idx, :]$

end parallel For

Store $\mathbf{Acc}$ to global memory as $\mathbf{A}_{time}$.

J Proofs of Propositions in §3.2

Theorem 1. *Consider a transformer with hidden dimension* $2c$*. Let* $x_{t,k}$ *denote the pre-RoPE key value at token position* t *and channel* k*, where* $0 \le t \le G-1$ *and* $0 \le k \le 2c-1$*.*

Define the complex-valued representation

$$z_{t,j} := x_{t,2j} + i\, x_{t,2j+1}, \tag{14}$$

and apply Rotary Position Embedding (RoPE) in the complex plane:

$$w_{t,j} := e^{it\theta_j} z_{t,j}, \qquad \theta_j = b^{-j/c}, \tag{15}$$

where i *is the imaginary unit.*

Let $\mathrm{DCT}(\cdot)$ *denote the Type-II Discrete Cosine Transform along the time dimension* t*. Applying it to post-RoPE key value produces the spectral coefficients* $c_{k,j}$ *for frequency indices* $0 \le k \le G-1$*.*

Then the following properties hold:

1. ***Dominant Frequency*** *If* $z_{t,j} = z_{0,j}$ *for all* t*, there exists a* G_0*, for all* $G > G_0$*, the DCT coefficient peaks at one of the two integer frequency indices nearest to* $\theta_j G/\pi$*:*

$$\operatorname*{argmax}_k |c_{k,j}| \in \left\{ \left\lfloor \frac{\theta_j G}{\pi} \right\rfloor, \left\lceil \frac{\theta_j G}{\pi} \right\rceil \right\}. \tag{16}$$

2. ***Energy concentration.*** *Under the same position-invariant assumption, the total spectral energy satisfies*

$$\sum_{k=0}^{G-1} |c_{k,j}|^2 = \sum_{t=0}^{G-1} |w_{t,j}|^2 = |z_{0,j}|^2\, G \;=: E_j. \tag{17}$$

Let $k_0 = \lfloor \theta_j G/\pi \rfloor$ *and* $k_1 = \lceil \theta_j G/\pi \rceil$ *be the two nearest integer frequency indices. Then at least 60% of the energy is concentrated in these two bins:*

$$|c_{k_0,j}|^2 + |c_{k_1,j}|^2 \ge \frac{6}{\pi^2} E_j. \tag{18}$$

3. ***Stability of Spectral Coefficients.*** *Let the semantic signal be decomposed into a constant mean and a time-varying fluctuation:* $z_{t,j} = \bar{z}_j + \delta_{t,j}$*. The mean squared error (MSE) between the DCT spectrum of the actual signal and the spectrum of the ideal RoPE carrier is tightly bounded by the variance of the semantic signal:*

$$\frac{1}{G} \sum_{k=0}^{G-1} \left| c_{k,j} - c_{k,j}^{(ideal)} \right|^2 = \mathrm{Var}(z_{:,j}), \tag{19}$$

where $c_{k,j}^{(ideal)} = \mathrm{DCT}\{e^{it\theta_j} \bar{z}_j\}_k$*. This implies that for smoothly varying semantic signals (where* $\mathrm{Var}(z) \ll |\bar{z}|^2$*), the spectral energy distribution remains dominated by the dominant frequency derived in Property 1.*

Proof. **Proof of statement 1 (Dominant Frequency).** Without loss of generality, we assume $z_{0,j} = 1$. By taking DCT transform, we have

$$c_{k,j} = \alpha_k \sum_{t=0}^{G-1} e^{i\theta_j t} \cos[\frac{\pi k}{G}(t + \frac{1}{2})].$$

It follows from

$$\cos[\frac{\pi k}{G}(t + \frac{1}{2})] = \frac{1}{2}[e^{i\frac{\pi k}{G}(t+\frac{1}{2})} + e^{-i\frac{\pi k}{G}(t+\frac{1}{2})}]$$

that

$$c_{k,j} = \frac{\alpha_k}{2} \sum_{t=0}^{G-1} [e^{i\theta_j t} e^{i\frac{\pi k}{G}(t+\frac{1}{2})} + e^{i\theta_j t} e^{-i\frac{\pi k}{G}(t+\frac{1}{2})}]$$
$$= \frac{\alpha_k}{2} e^{i\frac{\pi k}{2G}} \sum_{t=0}^{G-1} e^{it(\theta_j + \frac{\pi k}{G})} + e^{-i\frac{\pi k}{2G}} \sum_{t=0}^{G-1} e^{it(\theta_j - \frac{\pi k}{G})}.$$

We denote

$$\phi_1 = \theta_j + \frac{\pi k}{G}, \quad \phi_2 = \theta_j - \frac{\pi k}{G}, \tag{20}$$

then

$$S_1 := \sum_{t=0}^{G-1} e^{it(\theta_j + \frac{\pi k}{G})} = \frac{1 - e^{iG\phi_1}}{1 - e^{i\phi_1}}.$$

Similarly,

$$S_2 := \sum_{t=0}^{G-1} e^{it\phi_2} = \frac{1 - e^{iG\phi_2}}{1 - e^{i\phi_2}}.$$

Using the identity for a finite geometric series, we can simplify S_1 and S_2 as follows. Note that

$$1 - e^{iG\phi_1} = e^{iG\phi_1/2} \left(e^{-iG\phi_1/2} - e^{iG\phi_1/2} \right) = e^{iG\phi_1/2} \cdot \left[-2i \sin(G\phi_1/2) \right], \tag{21}$$

and

$$1 - e^{i\phi_1} = e^{i\phi_1/2} \cdot \left[-2i \sin(\phi_1/2) \right]. \tag{22}$$

Therefore,

$$S_1 = e^{i(G-1)\phi_1/2} \frac{\sin(G\phi_1/2)}{\sin(\phi_1/2)}.$$

Similarly,

$$S_2 = e^{i(G-1)\phi_2/2} \frac{\sin(G\phi_2/2)}{\sin(\phi_2/2)}.$$

Substituting back, we obtain

$$c_{k,j} = \frac{\alpha_k}{2} \left(e^{i\frac{\pi k}{2G}} S_1 + e^{-i\frac{\pi k}{2G}} S_2 \right)$$
$$= \frac{\alpha_k}{2} \left(e^{i\frac{\pi k}{2G}} e^{i(G-1)\phi_1/2} \frac{\sin(G\phi_1/2)}{\sin(\phi_1/2)} + e^{-i\frac{\pi k}{2G}} e^{i(G-1)\phi_2/2} \frac{\sin(G\phi_2/2)}{\sin(\phi_2/2)} \right)$$
$$= \frac{\alpha_k}{2} e^{i(G-1)\theta_j/2} \left(e^{i\frac{\pi k}{2}} \frac{\sin(\frac{\theta_j G + \pi k}{2})}{\sin(\frac{\theta_j}{2} + \frac{\pi k}{2G})} + e^{-i\frac{\pi k}{2}} \frac{\sin(\frac{\theta_j G - \pi k}{2})}{\sin(\frac{\theta_j}{2} - \frac{\pi k}{2G})} \right).$$

By taking the modulus of $c_{k,j}$, we have

$$|c_{k,j}| = \frac{\alpha_k}{2} \left| e^{i\frac{\pi k}{2}} \frac{\sin(\frac{\theta_j G + \pi k}{2})}{\sin(\frac{\theta_j}{2} + \frac{\pi k}{2G})} + e^{-i\frac{\pi k}{2}} \frac{\sin(\frac{\theta_j G - \pi k}{2})}{\sin(\frac{\theta_j}{2} - \frac{\pi k}{2G})} \right|$$
$$= \frac{\alpha_k}{2} \left| \sin(\frac{\theta_j G + \pi k}{2}) \right| \left| \frac{1}{\sin(\frac{\theta_j}{2} + \frac{\pi k}{2G})} + \frac{1}{\sin(\frac{\theta_j}{2} - \frac{\pi k}{2G})} \right|$$
$$:= \frac{\alpha_k}{2} \beta_k \left| \frac{1}{\sin(\frac{\theta_j}{2} + \frac{\pi k}{2G})} + \frac{1}{\sin(\frac{\theta_j}{2} - \frac{\pi k}{2G})} \right|.$$

Let

$$\overline{\alpha} = \max_k \alpha_k, \quad \underline{\alpha} = \min_k \alpha_k,$$
$$\overline{\beta} = \max_k \beta_k, \quad \underline{\beta} = \min_k \beta_k,$$

Case 1: $|\frac{\theta_j G}{\pi} - k| \leq \frac{1}{2}$ **(the nearest integer).**

$$|c_{k,j}| \geq \underline{\alpha}\,\underline{\beta}\left(\left|\frac{1}{\sin(\frac{\theta_j}{2} - \frac{\pi k}{2G})}\right| - \left|\frac{1}{\sin(\frac{\theta_j}{2} + \frac{\pi k}{2G})}\right|\right) \tag{23}$$
$$\geq \underline{\alpha}\,\underline{\beta}\left(\frac{1}{\sin(\frac{\pi}{2G})} - 1\right). \tag{24}$$

Case 2: $|\frac{\theta_j G}{\pi} - k| \geq 1$ **(the integers away from dominant frequency).**

$$|c_{k,j}| \leq \overline{\alpha}\,\overline{\beta}\left(\left|\frac{1}{\sin(\frac{\theta_j}{2} - \frac{\pi k}{2G})}\right| + \left|\frac{1}{\sin(\frac{\theta_j}{2} + \frac{\pi k}{2G})}\right|\right)$$
$$\leq \overline{\alpha}\,\overline{\beta}\left(\frac{1}{\sin(\frac{\pi}{G})} + \max\left\{\left|\frac{1}{\sin(\frac{\theta_j}{2})}\right|, \left|\frac{1}{\sin(\frac{\theta_j}{2} + \frac{\pi(G-1)}{2G})}\right|\right\}\right).$$

Now we determine how large G needs to be so that the lower bound in Case 1 exceeds the upper bound in Case 2. Using the inequalities $\sin x \geq \frac{2}{\pi}x$ for $0 \leq x \leq \frac{\pi}{2}$ and $\sin x \leq x$ for $x \geq 0$, we obtain

$$\frac{1}{\sin(\frac{\pi}{2G})} \geq \frac{2G}{\pi}, \qquad \frac{1}{\sin(\frac{\pi}{G})} \leq \frac{G}{2}. \tag{25}$$

Let

$$M := \max\Big\{\Big|\frac{1}{\sin(\frac{\theta_j}{2})}\Big|,\ \Big|\frac{1}{\sin(\frac{\theta_j}{2} + \frac{\pi(G-1)}{2G})}\Big|\Big\}. \tag{26}$$

For large G, the second argument tends to $\frac{\theta_j}{2} + \frac{\pi}{2}$, hence $M \leq C_\theta$ where $C_\theta := \max\big\{|\csc(\frac{\theta_j}{2})|,\ |\sec(\frac{\theta_j}{2})|\big\}$. Using the notation $\underline{a} = \underline{\alpha}\underline{\beta}$ and $\overline{a} = \overline{\alpha}\overline{\beta}$, the required condition becomes

$$\underline{a}\Big(\frac{2G}{\pi} - 1\Big) > \overline{a}\Big(C_\theta + \frac{G}{2}\Big). \tag{27}$$

Solving for G gives

$$G > \frac{\pi\big(\overline{a}C_\theta + \underline{a}\big)}{2\underline{a} - \frac{\pi}{2}\overline{a}} = \frac{\pi\big(\overline{\alpha}\overline{\beta}C_\theta + \underline{\alpha}\underline{\beta}\big)}{2\underline{\alpha}\underline{\beta} - \frac{\pi}{2}\overline{\alpha}\overline{\beta}}. \tag{28}$$

This completes the proof of the first statement of the theorem.

Proof of statement 2 (Energy concentration). First, the Parseval identity for the DCT-II transform gives

$$\sum_{k=0}^{G-1} |c_{k,j}|^2 = \sum_{t=0}^{G-1} |w_{t,j}|^2. \tag{29}$$

Without loss of generality, we assume $z_{t,j} = z_{0,j} = 1$ for all t, we have $w_{t,j} = e^{i\theta_j t}$ and therefore $|w_{t,j}| = 1$. Hence

$$\sum_{t=0}^{G-1} |w_{t,j}|^2 = G = |z_{0,j}|^2\, G =: E_j. \tag{30}$$

Thus the total energy E_j is preserved by the DCT.

Let $k_0 = \lfloor \theta_j G/\pi \rfloor$ and $k_1 = \lceil \theta_j G/\pi \rceil$ be the two integers nearest to $\theta_j G/\pi$. Without loss of generality, we assume $|k_0 - \theta_j G/\pi| \leq \frac{1}{2}$. Applying (24), we can deduce

$$\begin{aligned}
|c_{k_0,j}|^2 \geq & \frac{1}{G} \left| \sin(\frac{\theta_j G + \pi k_0}{2}) \right|^2 \left(\frac{1}{\sin(\frac{\pi}{2G})} - 1 \right)^2 \\
\geq & \frac{1}{G} \left| \sin(\frac{\theta_j G + \pi k_0}{2}) \right|^2 \left(\frac{2G}{\pi} - 1 \right)^2 .
\end{aligned}$$

For the integer k_1 adjacent to k_0, we have

$$\begin{aligned}
|c_{k_1,j}|^2 \geq & \frac{2}{G} \left| \sin(\frac{\theta_j G + \pi k_1}{2}) \right|^2 \left(\frac{1}{\sin(\frac{\pi}{G})} - 1 \right)^2 \\
\geq & \frac{2}{G} \left| \sin(\frac{\theta_j G + \pi k_1}{2}) \right|^2 \left(\frac{G}{\pi} - 1 \right)^2 \\
= & \frac{2}{G} \left| \cos(\frac{\theta_j G + \pi k_0}{2}) \right|^2 \left(\frac{G}{\pi} - 1 \right)^2 .
\end{aligned}$$

Then

$$\begin{aligned}
|c_{k_0,j}|^2 + |c_{k_1,j}|^2 \geq & \frac{1}{G} \left(\frac{2G}{\pi} - 1 \right)^2 + \left(\frac{G}{\pi} - 1 \right)^2 \\
\geq & \frac{6G}{\pi^2} - \frac{8}{\pi} + \frac{3}{G} \\
\approx & \frac{6G}{\pi^2}.
\end{aligned}$$

Thus

$$|c_{k_0,j}|^2 + |c_{k_1,j}|^2 \geq \frac{6}{\pi^2} E_j, \tag{31}$$

completing the proof of the second statement.

Proof of statement 3 (Stability of Spectral Coefficients). We now consider general sequences $\{z_{t,j}\}_{t=0}^{G-1}$ that may vary with time. Let $w_{t,j} = e^{i\theta_j t} z_{t,j}$ and $\widetilde{w}_{t,j} = e^{i\theta_j t} \overline{z}_j$. Denote by $\hat{w}_{k,j}$ and $\hat{\widetilde{w}}_{k,j}$ their respective DCT-II coefficients, i.e. $\hat{w}_{k,j} = c_{k,j}$ and $\hat{\widetilde{w}}_{k,j}$ is the DCT of $\widetilde{w}_{t,j}$. By the Parseval identity already used above, we have

$$\sum_{k=0}^{G-1} |\hat{w}_{k,j} - \hat{\widetilde{w}}_{k,j}|^2 = \sum_{t=0}^{G-1} |w_{t,j} - \widetilde{w}_{t,j}|^2 = \sum_{t=0}^{G-1} |e^{i\theta_j t}(z_{t,j} - \overline{z}_j)|^2 = \sum_{t=0}^{G-1} |z_{t,j} - \overline{z}_j|^2. \tag{32}$$

Thus the proof of the third statement is complete.

□