**Preprint – IJCAI 2025**: This is the accepted version made available for conference attendees.
**Do not cite**. The final version will appear in the IJCAI 2025 proceedings.

# RotateKV: Accurate and Robust 2-Bit KV Cache Quantization for LLMs via Outlier-Aware Adaptive Rotations

**Zunhai Su**[1], **Hanyu Wei**[1], **Zhe Chen**[2], **Wang Shen**[2], **Linge Li**[2], **Huangqi Yu**[2] and **Kehong Yuan**[*1]

[1]Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
[2]Huawei Technologies Co., Ltd., Shenzhen, China
{zh-su23,wei-hy23}@mails.tsinghua.edu.cn, {chenzhe49,shenwang1,lilinge,yuhuangqi}@huawei.com, yuankh@sz.tsinghua.edu.cn

## Abstract

Key-Value (KV) cache facilitates efficient large language models (LLMs) inference by avoiding recomputation of past KVs. As the batch size and context length increase, the oversized KV caches become a significant memory bottleneck, highlighting the need for efficient compression. Existing KV quantization rely on fine-grained quantization or the retention of a significant portion of high bit-widths caches, both of which compromise compression ratio and often fail to maintain robustness at extremely low average bit-widths. In this work, we explore the potential of rotation technique for 2-bit KV quantization and propose RotateKV, which achieves accurate and robust performance through the following innovations: (i) Outlier-Aware Rotation, which utilizes channel-reordering to adapt the rotations to varying channel-wise outlier distributions without sacrificing the computational efficiency of the fast Walsh-Hadamard transform (FWHT); (ii) Pre-RoPE Grouped-Head Rotation, which mitigates the impact of rotary position embedding (RoPE) on proposed outlier-aware rotation and further smooths outliers across heads; (iii) Attention-Sink-Aware Quantization, which leverages the massive activations to precisely identify and protect attention sinks. RotateKV achieves less than 0.3 perplexity (PPL) degradation with 2-bit quantization on WikiText-2 using LLaMA-2-13B, maintains strong CoT reasoning and long-context capabilities, with less than 1.7% degradation on GSM8K, outperforming existing methods even at lower average bit-widths. RotateKV also showcases a 3.97× reduction in peak memory usage, supports 5.75× larger batch sizes, and achieves a 2.32× speedup in decoding stage.

## 1 Introduction

Large language models (LLMs) have attracted considerable attention due to their remarkable capabilities in next-token-prediction generation tasks [Zhao *et al.*, 2023]. A critical technique of LLMs is the Key-Value (KV) cache, which avoids recomputation by caching the KVs generated by the attention layer in each Transformer block [Vaswani, 2017]. However, as the batch size and context length increase, the growing size of KV caches not only leads to significant memory consumption, but also renders LLM inference memory-bound, limiting system throughput [Liu *et al.*, 2024d; Kang *et al.*, 2024]. Low-bit quantization is widely used to compress LLMs for enhanced memory and time efficiency, covering various aspects, including weight-only quantization [Frantar *et al.*, 2022; Lin *et al.*, 2024b], weight-activation quantization [Xiao *et al.*, 2023a; Ashkboos *et al.*, 2024; Liu *et al.*, 2024c; Saxena *et al.*, 2024], and KV cache quantization. Existing studies on KV cache quantization often employ techniques like fine-grained per-channel quantization [Liu *et al.*, 2024d; Kang *et al.*, 2024] or mixed-precision quantization [He *et al.*, 2024; Yang *et al.*, 2024; Duanmu *et al.*, 2024; Hooper *et al.*, 2024], which retain a significant portion of high bit-widths caches. As a result, these methods compromise compression efficiency and fail to maintain robustness under high compression ratio or extremely low average bit-widths.

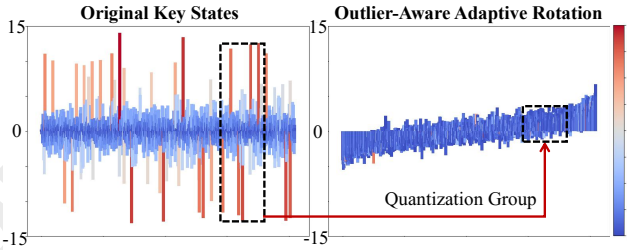
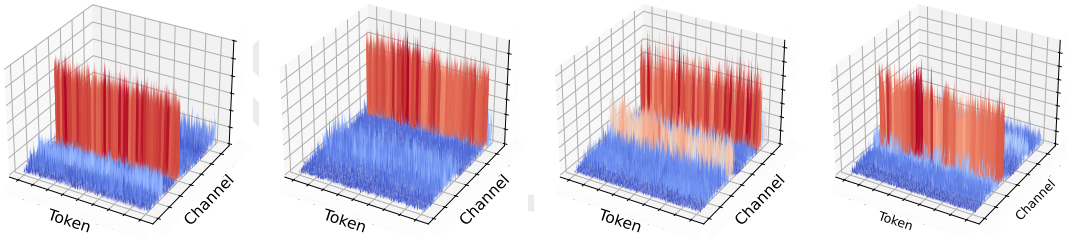



Figure 1: The distribution of Keys in Layer 10 of LLaMA-2-7B. The proposed outlier-aware adaptive rotation demonstrates outstanding capability in reducing outliers.

Recently, Hadamard-transform-based rotation technique has demonstrated significant effectiveness in mitigating outliers in 4-bit LLM quantization, with studies such as [Ashkboos *et al.*, 2024; Liu *et al.*, 2024c; Saxena *et al.*, 2024] leveraging rotation technique to achieve 4-bit quantization of weights, activations and KV caches. However, the potential of rotation methods for extremely low-bit KV cache quantization has yet to be fully explored. In this study, we thoroughly analyze the limitations of existing KV rotation settings and

*Corresponding author: Kehong Yuan

**Preprint – IJCAI 2025**: This is the accepted version made available for conference attendees.
**Do not cite**. The final version will appear in the IJCAI 2025 proceedings.




(a) Layer 10 Head 1 (b) Layer 10 Head 2 (c) Layer 10 Head 30 (d) Layer 10 Head 31

Figure 2: Magnitude of of LLaMA-2-7B Keys. A small number of channels exhibit disproportionately large magnitudes, and these outlier channels vary across different attention heads.

propose RotateKV, a tuning-free method that ensures superior outlier management through the following innovations (see Figure 1 for an illustrative example):

**Outlier-Aware Rotation.** Recent research [Liu *et al.*, 2024d; Hooper *et al.*, 2024] on outlier behavior in KV cache highlight the Keys contain a small number of channels with disproportionately large magnitudes, while these outlier channels vary across different attention heads, as shown in Figure 2. However, existing Key rotation rely on the fast Walsh-Hadamard transform (FWHT), where the same Hadamard matrix is applied to each head across all layers. Although the FWHT reduces the overhead of online computation, its rotation fails to adapt to the varying distributions of channel-wise outliers. Through a comprehensive comparative analysis of various outlier-aware strategies, we propose outlier-aware rotation, which effectively enhances the adaptability of rotation to diverse outlier distributions while maintaining the computational efficiency of FWHT.

**Pre-RoPE Grouped-Head Rotation.** Existing rotations of Keys are performed by applying the rotation to each head of the Query and Key after rotary position embedding (RoPE) [Su *et al.*, 2024], which presents two significant limitations that reduce the effectiveness of outlier-aware rotation in mitigating outliers. First, RoPE disrupts the channel magnitude consistency of the Key tensors. Second, applying the rotation through the attention calculation of each head confines outlier reduction to individual heads. To address these issues, we propose a novel pre-RoPE rotation pipeline, which reduces the impact of RoPE on rotation and enables grouped-head rotation to smooth outliers across multiple heads.

**Attention-Sink-Aware Quantization.** Retaining attention sinks [Xiao *et al.*, 2023b] is crucial for maintaining model performance during compression [Duanmu *et al.*, 2024; Hooper *et al.*, 2024; Liu *et al.*, 2024b]. Existing methods primarily focus on retaining sink tokens at the beginning of sequences, overlooking those in other positions [Yu *et al.*, 2024; Sun *et al.*, 2024]. Inspired by recent studies on the interpretability of attention sinks [Sun *et al.*, 2024; Guo *et al.*, 2024], we propose attention-sink-aware quantization, which leverages the correlation between massive activations [Sun *et al.*, 2024] and attention sinks to precisely identify and retain additional salient sink tokens.

With these innovations, RotateKV fully exploits the potential of rotation for extremely low-bit quantization, addressing the limitations of current KV quantization methods and delivering both accurate and robust 2-bit quantization. **Our contributions are summarized as follows:**

- We identify the limitation of existing Key rotation that applies the same rotation to all attention heads. We propose outlier-aware rotation, which improves adaptability to diverse outlier distributions, while maintaining the computational efficiency of FWHT.
- Building on our analysis of the impact and limitations of existing post-RoPE rotations, we propose pre-RoPE rotation that effectively mitigates RoPE's influence on rotations while unlocking the potential for grouped-head rotation.
- We propose attention-sink-aware quantization, an innovative approach that provides new insights into identifying attention sinks and extends the current practice of retaining only the sink tokens at the beginning of the sequence.
- To the best of our knowledge, RotateKV is the first method to fully explore and harness the potential of rotation in extremely low-bit KV quantization. Extensive experiments demonstrate that RotateKV outperforms state-of-the-art methods while achieving higher compression ratio. RotateKV achieves less than 0.3 perplexity (PPL) degradation with 2-bit on WikiText-2 [Merity *et al.*, 2016] using LLaMA-2-13B [Touvron *et al.*, 2023]. Evaluations on the GSM8K [Cobbe *et al.*, 2021] demonstrates that RotateKV maintains strong chain-of-thought (CoT) reasoning capability, with less than 1.7% performance degradation compared to FP16 baseline. RotateKV also exhibits negligible performance degradation in the 40K context-length Needle-in-a-Haystack (NIAH) [Fu *et al.*, 2025] test, as well as in eight challenging long-context and multi-modal tasks chosen from LongBench [Bai *et al.*, 2023b] and MileBench [Song *et al.*, 2024]. With the current implementation, RotateKV can reduce peak memory usage by 3.97×, supports 5.75× larger batch sizes, and achieves a 2.32× decoding speedup.

## 2 Related Work

### 2.1 KV Cache Quantization

Quantization reduces the bit-widths of numerical representation to compress the KV cache, effectively decreasing memory usage and alleviating the memory-bound challenge. Generally, existing KV cache quantization methods can be cat-

**Preprint – IJCAI 2025**: This is the accepted version made available for conference attendees.
**Do not cite**. The final version will appear in the IJCAI 2025 proceedings.

egorized into two types based on the quantization dimension of Keys: per-channel and per-token. Due to significant outliers along the channel dimension in Keys, methods such as KIVI [Liu *et al.*, 2024d], Gear [Kang *et al.*, 2024], and KVQuant [Hooper *et al.*, 2024] adopt per-channel quantization to mitigate quantization errors. However, these approaches often require fine-grained quantization or the retention of a certain proportion of outliers unquantized to preserve model performance, both of which compromise the compression ratio. On the other hand, due to the autoregressive nature of LLMs, where tokens are predicted sequentially, per-token quantization is commonly used in methods such as ZipCache [He *et al.*, 2024], MiKV [Yang *et al.*, 2024], SKVQ [Duanmu *et al.*, 2024], and other less aggressive KV quantization methods [Ashkboos *et al.*, 2024; Liu *et al.*, 2024c; Shah *et al.*, 2024]. These approaches typically focus on the saliency differences between tokens, allocating relatively higher bit-widths to store a significant proportion of salient tokens. However, this also results in a compromised compression ratio. Unlike these methods, RotateKV employs per-token quantization but eliminates the need to store large proportions of high-bit caches. Besides, experiments show that RotateKV offers superior outlier management compared to per-channel approaches, ensuring robust performance even at higher compression ratio.

# 3 Preliminary

## 3.1 Quantization

The n-bit asymmetric integer quantization and dequantization processes can be expressed as:

$$Q(X) = clamp\left(\left\lfloor \frac{X}{scale} \right\rceil + zero, 0, 2^n - 1\right), \quad (1)$$

$$X' = scale \cdot (Q(X) - zero), \quad (2)$$

$$scale = \frac{clipped_max(X) - clipped_min(X)}{2^n - 1}, \quad (3)$$

$$zero = -\left\lfloor \frac{clipped_min(X)}{scale} \right\rceil, \quad (4)$$

where $\lfloor \cdot \rceil$ indicates round operation. $Q(X)$ and $X'$ denote the quantized and dequantized values of $X$. The $clamp$ operation constrains the values within a specified range. $clipped_max(X)$ and $clipped_min(X)$ denote the operations that truncate the maximum and minimum values of $X$.

## 3.2 Hadamard Matrix

Hadamard Matrix $R$ is a specific type of orthogonal matrix characterized by entries proportional to $\{+1, -1\}$. The Hadamard matrix also satisfies the definition of a rotation matrix, as it is an orthogonal matrix with $\det(R) = 1$. QuIP [Chee *et al.*, 2024] demonstrates that multiplying an rotation matrix leads to a reduction in the maximum entry relative to its norm, effectively mitigating outliers and enhancing quantizability. Walsh-Hadamard Matrix is a particular instance of Hadamard matrix generated recursively as follows, the subscript denoting the dimension of matrix, where $k \in \mathbb{Z}^+$:

$$H_1 = [1], \quad H_{2^k} = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{2^{(k-1)}} & H_{2^{(k-1)}} \\ H_{2^{(k-1)}} & -H_{2^{(k-1)}} \end{bmatrix}. \quad (5)$$

The scaling factor $\frac{1}{\sqrt{2}}$ ensures normalization. The recursive structure of the Walsh-Hadamard matrix enables efficient computation via the FWHT algorithm [Hedayat and Wallis, 1978], reducing computational complexity of matrix-vector multiplication to $O(n \log n)$, where $n$ represents the dimension of the matrix.

# 4 Methodology

Section 4.1 outlines the limitations of existing outlier-unaware Key rotation and proposes outlier-aware rotation. In Section 4.2, we analyze the impact of RoPE on rotation, then propose pre-RoPE grouped-head rotation. Section 4.3 further introduces attention-sink-aware quantization. Section 4.4 provides a comprehensive summary of RotateKV. The overview of RotateKV is illustrated in Figure 3.

## 4.1 Outlier-Aware Rotation

**Existing Outlier-Unaware Key Rotation**

When applying quantization, outliers pose a persistent challenge as they expand the quantization range, reducing the effective bits available for most values. Recently, LLM quantization research [Chee *et al.*, 2024; Ashkboos *et al.*, 2024; Liu *et al.*, 2024c; Saxena *et al.*, 2024; Lin *et al.*, 2024a; Shah *et al.*, 2024] has focused on Hadamard-transform-based rotation technique, which involves multiplying a Hadamard matrix to reduce outliers and improve quantizability. As shown in Figure 4, existing rotations within the attention layer can be classified into two categories: offline rotations (e.g., $R_1$ and $R_2$), which can be fused into the weights prior to inference to eliminate the overhead of online computations, and online rotations (e.g., $R_3$), which are performed dynamically using the FWHT. This differentiation exists because the Query and Key computation relies on RoPE, rendering it incompatible with offline rotations. To reduce the computational overhead during inference, Key rotations typically utilize the efficient FWHT algorithm. For example, QuaRot [Ashkboos *et al.*, 2024] leverages offline random Hadamard matrices and online FWHT to achieve outlier-free 4-bit LLM inference. SpinQuant [Liu *et al.*, 2024c] employs Cayley optimization to enhance offline rotation but continues to utilize the FWHT for online rotations. Due to the structure of the Walsh-Hadamard matrix, as shown in Equation 5, the rotations for each attention head with the same dimension rely on the same Walsh-Hadamard matrix. This constraint limits adaptability to varying channel-wise outlier distributions.

**Outlier-Aware Rotation**

While adjusting rotation for pre-rotation distributions seems straightforward, it is impractical. Modifying the Hadamard matrices invalidates the FWHT algorithm, increases computational overhead, and necessitates storing multiple matrices, thereby further reducing compression ratio. To adapt rotations to varying channel-wise outlier distributions without compromising the efficiency of FWHT, we introduce the outlier-aware rotation, which preserves the FWHT while enhancing it with outlier-aware operation derived from lightweight calibration. We conduct experiments using various enhancement strategies, including smoothing and reordering. Smoothing [Xiao *et al.*, 2023a; Lin *et al.*, 2024c]

**Preprint – IJCAI 2025**: This is the accepted version made available for conference attendees.
**Do not cite**. The final version will appear in the IJCAI 2025 proceedings.

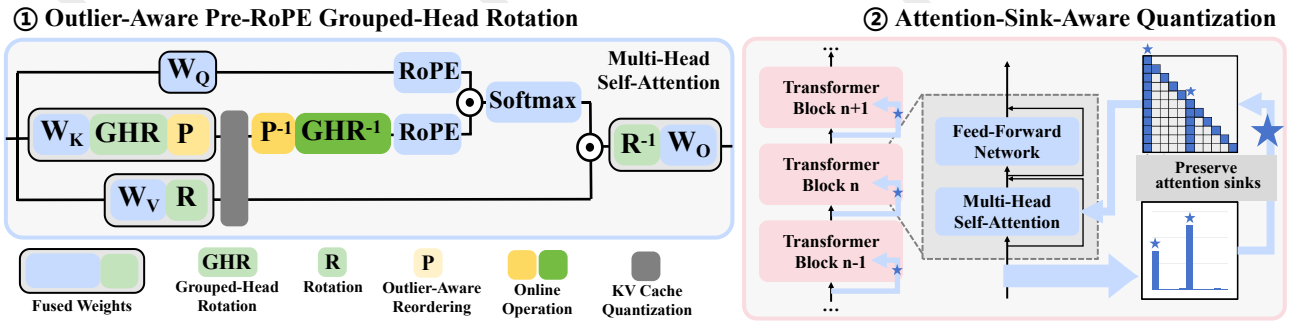



Figure 3: Overview of RotateKV. On the left is the outlier-aware rotation combined with the pre-RoPE pipeline. On the right, we demonstrate attention-sink-aware quantization. Since attention is concentrated on the massive activations, we can identify attention sinks in the current attention layer by utilizing the token indices of these massive activations from the output of the previous decoder block.

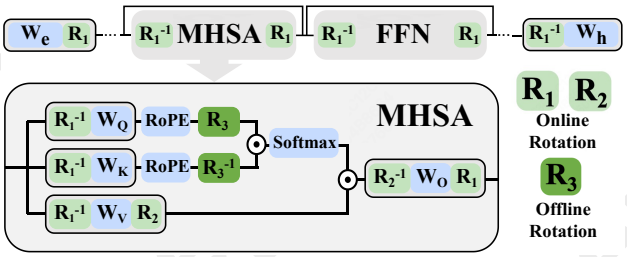



Figure 4: Existing rotation paradigm.

| Methods | LLaMA-2-7B PPL ↓ | | | |
|---|---|---|---|---|
| | 16bit | 4bit | 3bit | 2bit |
| Rotate-Only | 5.47 | 5.62 | 6.16 | 8.05 |
| Smooth-Only | | 5.71 | 6.65 | 16.97 |
| Reorder-Only | | 5.73 | 6.47 | 8.26 |
| Rotate + Smooth | | 5.62 | 6.13 | 6.93 |
| Smooth + Rotate | | 5.61 | 6.02 | 6.83 |
| **Rotate + Reorder** | | **5.63** | **6.07** | **6.33** |
| Reorder + Rotate | | 7.43 | 36.94 | 231.43 |

Table 1: Experiments on different outlier-aware strategies. We evaluate PPL with a sequence length of 2048 using the LLaMA-2-7B model on the WikiText-2 dataset.

scales down the Keys by per-channel factors $\lambda$ and correspondingly scaling up the associated channels in the Query:

$$\mathbf{Q} \cdot \mathbf{K}^T = (\mathbf{Q}\mathbf{\Lambda}) \cdot (\mathbf{K}\mathbf{\Lambda}^{-1})^T, \quad \mathbf{\Lambda} = \text{diag}(\lambda). \tag{6}$$

Reordering arranges channels across all heads to reduce outliers within each quantization group. As shown in Table 1, experiments on different outlier-aware strategies demonstrate that smoothing fails to maintain performance at 2-bit quantization. For reordering, we find that relying on clustering or other complex permutation methods is unnecessary. Instead, reordering the channels of each token using the indices sorted by the values after rotation can effectively reduce quantization errors (Table 2) and improve PPL. Notably, the channel reordering indices are derived through fast calibration and remain consistent across all tokens throughout inference. The calibration procedure is detailed in Algorithm 1.

**Algorithm 1:** Calibration for Reordering Indices

**Input**: Key states after rotation: $K \in \mathbb{R}^{b \times h \times s \times d}$
**Parameter**: Number of layers: $L$, batch size: $b$, number of attention heads: $h$, sequence length: $s$, dimensionality of each head: $d$.
**Output**: Reordering indices $P_l$ for each layer $l \in \{1, 2, \ldots, L\}$

**for** $l = 1$ **to** $L$ **do**
  Reshape $K \in \mathbb{R}^{bs \times hd}$.
  Compute $channel_sum^l = \sum_{i=1}^{bs} K[i, :]$.
  Sort indices: $P_l = \text{argsort}(channel_sum^l)$.
**end for**
**return** $P_l$ for each $l \in \{1, 2, \ldots, L\}$.

| Methods | MSE ↓ | |
|---|---|---|
| | Pre-RoPE | Post-RoPE |
| Rotation | 0.632 | 0.670 |
| **+ Reordering** | **0.247** | **0.605** |
| Reordering | 0.477 | 0.568 |
| + Rotation | NaN | NaN |

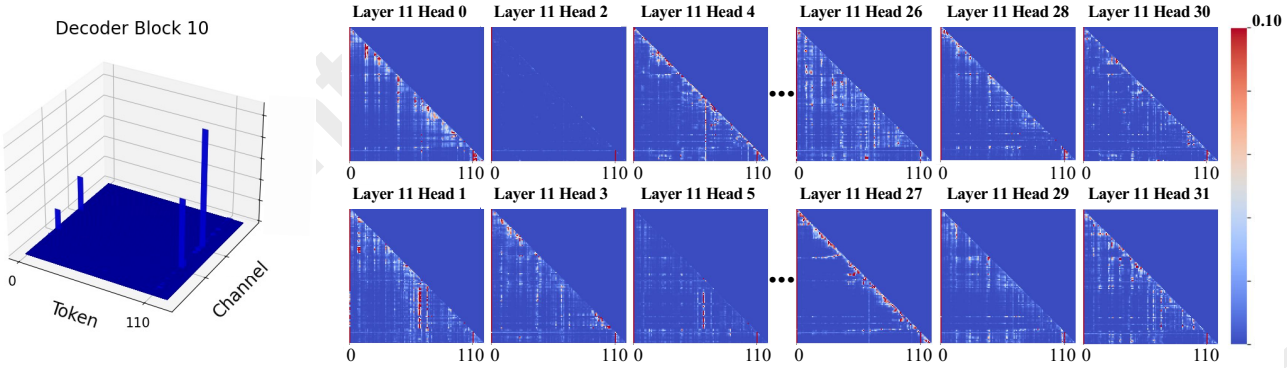
Table 2: We assess the reconstruction error of the Key in LLaMA-2-7B under 2-bit quantization, using mean squared error (MSE) as the metric. The results indicate that RoPE significantly affects the effectiveness of outlier-aware rotation.

## 4.2 Pre-RoPE Grouped-Head Rotation

### Existing Post-RoPE Key Rotation

Recent LLMs, including LLaMA [Touvron *et al.*, 2023], Mistral [Jiang *et al.*, 2023] and Qwen [Bai *et al.*, 2023a] use RoPE [Su *et al.*, 2024] to encode token positions. The positional encoding introduced by RoPE is applied to pairs of channels, resulting in a reduction of magnitude consistency within individual channel, as shown in Figure 6. Previous work [Hooper *et al.*, 2024] also highlights that RoPE affects per-channel quantization performance. In this study, we observe that the inconsistency in the magnitudes of channel-wise outliers significantly undermines the effectiveness of outlier-aware rotation. As demonstrated in Table 2, the quantization error increases by 145% after applying RoPE.

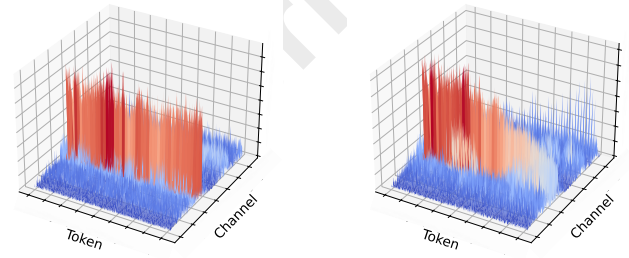
**Preprint – IJCAI 2025**: This is the accepted version made available for conference attendees.
**Do not cite**. The final version will appear in the IJCAI 2025 proceedings.




(a) Outputs of Decoder Block 10.

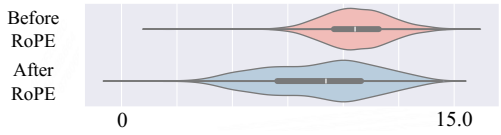
(b) In this example, attention sinks are observed at tokens 0 and 110.

Figure 5: Visualizations of Decoder Block 10 outputs and the attention scores in Attention Layer 11 from the LLaMA-2-7B using input from WikiText-2 dataset. As shown in Figure 5a, massive activations occur at tokens 0 and 110, in channels 1415 and 2533. In the subsequent attention layer, attention is focused on tokens 1 and 110 across all heads, as illustrated in Figure 5b.




(a) Key states before RoPE.

(b) Key states after RoPE.




(c) Distribution of Channel 51 before and after RoPE.

Figure 6: Figures 6a and 6b illustrate the Key of LLaMA-2-7B, Layer 10, Head 31, before and after RoPE respectively. Figure 6c presents the distribution of the outlier channel.

### Pre-RoPE Grouped-Head Rotation

To address the issues associated with post-RoPE rotation, we introduce a novel pipeline that applies the outlier-aware rotation before RoPE. As illustrated in Figure 3, this design not only eliminates RoPE's negative impact on rotation but also facilitates the incorporation of rotation and reordering operations into the weights, thus requiring only the inverse operation during inference. The proposed pre-RoPE rotation also enables grouped-head combined rotation, allowing for more effective outlier reduction across heads. Although increasing the head numbers improves PPL, it also incurs higher computational costs. Therefore, it is crucial to balance computational overhead and performance gains when determining the group size. Typically, as shown in Table 3, a group size of 4 heads is considered a reasonable choice.

| Heads per Group | Numbers of Group | Dimension of Rotation Matrix | FLOPs /Layer | LLaMA-2-7B PPL ↓ |
|---|---|---|---|---|
| 1 | 32 | 128 | 28672 | 7.81 |
| 2 | 16 | 256 | 32768 | 7.24 |
| **4** | **8** | **512** | **36864** | **6.99** |
| 8 | 4 | 1024 | 40960 | 6.91 |
| 16 | 2 | 2048 | 45056 | 6.98 |
| 32 | 1 | 4096 | 49152 | 6.86 |

Table 3: Experiments on group size of grouped-head rotation. The PPL is evaluated on the LLaMA-2-7B using WikiText-2 dataset.

### 4.3 Attention-Sink-Aware Quantization

The research [Xiao *et al.*, 2023b] indicates that LLMs tend to treat the initial token as a 'sink', assigning it disproportionately high attention scores. Moreover, [Hooper *et al.*, 2024; Duanmu *et al.*, 2024; Liu *et al.*, 2024b] highlight that the KVs associated with sink tokens are sensitive to quantization, and retaining only the initial token in FP16 can effectively enhance quantization. However, more recent studies [Yu *et al.*, 2024; Sun *et al.*, 2024] suggest that these few-in-number attention sinks can emerge not only at the initial token but also at various other positions, whereas existing practices fail to retain them precisely. One key reason is that efficient attention computation relies on kernels such as FlashAttention [Shah *et al.*, 2024], which directly output the attention results without exposing the intermediate attention scores. This prevents the dynamic identification of additional attention sinks beyond those fixed at the initial token. Inspired by recent studies on the interpretability of attention sinks [Sun *et al.*, 2024; Guo *et al.*, 2024], we propose attention-sink-aware quantization, which leverages massive activations to identify additional sink tokens without relying on attention scores, thereby precisely retaining them during the quantization process. Research on massive activations [Sun *et al.*, 2024]—those activations in the residual sums of Transformer block outputs with significantly larger magnitudes than others—suggests that attention is concentrated on these activations. Specifically, when massive activations occur, the corresponding tokens attract concentrated attention, forming atten-

**Preprint – IJCAI 2025**: This is the accepted version made available for conference attendees.
**Do not cite**. The final version will appear in the IJCAI 2025 proceedings.

| Methods | WikiText-2 PPL ↓ | | | |
|---|---|---|---|---|
| | LLaMA 2-7B | LLaMA 2-13B | LLaMA 3-8B | Mistral 7B |
| FP16 | 5.12 | 4.57 | 5.75 | 4.91 |
| QuaRot-4bit | 5.15 | 4.60 | 5.83 | 4.93 |
| KVQuant-4bit | 5.14 | 4.59 | 5.79 | 4.94 |
| **RotateKV-4bit** | **5.13** | **4.58** | **5.79** | **4.92** |
| QuaRot-3bit | 5.33 | 4.72 | 6.21 | 5.05 |
| KVQuant-3bit | 5.20 | 4.65 | 5.95 | 4.99 |
| **RotateKV-3bit** | **5.20** | **4.64** | **5.94** | **4.97** |
| QuaRot-2bit | 8.94 | 6.96 | 21.43 | 6.62 |
| KVQuant-2bit | 5.59 | 4.95 | 6.75 | 5.34 |
| **RotateKV-2bit** | **5.50** | **4.84** | **6.69** | **5.24** |

Table 4: PPL evaluations on WikiText-2 with a sequence length of 4096. For both QuaRot and our method, the quantization group size was set to 128. For KVQuant, we adopted the scheme of retaining 0.5% of numerical outliers in full precision during quantization.

tion sinks. Figure 5 provides a real example from LLaMA-2-7B [Touvron *et al.*, 2023]. Therefore, by identifying the massive activations, additional sink tokens can be pinpointed. The process of attention-sink-aware quantization is outlined in Figure 3.

### 4.4 Summary of RotateKV

In summary, RotateKV first performs fast calibration to obtain reordering indices, then integrates grouped-head rotation and channel reordering into the Key weights. These operations result in outlier-aware rotation of the Keys, making them more suitable for quantization. After updating the KV cache, inverse online reordering and rotation are applied. For the Value, we adopt a simple offline rotation as shown in Figure 3, since Values do not contain outliers like Keys.

## 5 Experiments

### 5.1 Experiment Settings

**Models and Tasks**

To validate the robustness of our method, we evaluate RotateKV on a variety of challenging tasks using both LLMs and visual-language models (VLMs), including LLaMA-2-7B/13B [Touvron *et al.*, 2023], LLaVA-v1.5-7B/13B [Liu *et al.*, 2024a], Mistral-7B [Jiang *et al.*, 2023], LLaVA-v1.6-Mistral-7B [Liu *et al.*, 2023], LLaMA-3-8B [Dubey *et al.*, 2024], and LLaMA-2-7B-80K [Fu *et al.*, 2024]. We begin by evaluating the PPL on the WikiText-2 dataset [Merity *et al.*, 2016]. Then, evaluation on the GSM8k dataset with CoT prompting is conducted to assess RotateKV's capability in handling complex CoT reasoning task. We further test long-context and multi-modal tasks accuracy across eight tasks from the LongBench [Bai *et al.*, 2023b] and MileBench [Song *et al.*, 2024].

**Quantization Settings**

We employ per-token asymmetric integer quantization for both Keys and Values, setting the quantization group size of RotateKV to 128 across all evaluations to demonstrate the accuracy and robustness of our method at relatively coarse quantization granularity. The group size for grouped-head rotation is consistently set to 4 across all the models and tasks we tested. We employ FP8 to store the scale parameters and INT8 for the zero-points, since this approach does not affect the results but improves the overall compression rate. The calibration process is highly efficient, taking less than five minutes on a single RTX 4090 GPU for the LLaMA-2-7B model using the WikiText-2 dataset. Additionally, the evaluation results demonstrate that calibration performed on WikiText-2 generalizes effectively to other datasets.

| Methods | GSM8K (8-shot) Accuracy ↑ | | | | Average Bits |
|---|---|---|---|---|---|
| | LLaMA 2-7B | LLaMA 2-13B | LLaMA 3-8B | Mistral 7B | |
| FP16 | 14.18 | 25.40 | 51.33 | 42.68 | 16 |
| KIVI | 13.19 | 24.64 | 43.44 | 39.12 | 2.50 |
| GEAR | 12.96 | 22.74 | 44.88 | 39.50 | 4.03 |
| MiKV | 9.02 | 21.00 | 30.93 | 36.47 | 3.23 |
| ZipCache | 13.50 | 25.02 | 49.20 | 41.32 | 3.23 |
| **RotateKV** | **13.95** | **25.09** | **50.49** | **42.99** | **2.25** |

Table 5: Evaluations on GSM8K with 8-shot CoT prompting.

### 5.2 Main Results and Analysis

**Perplexity Evaluation**

We use the original KV rotation and quantization scheme from QuaRot [Ashkboos *et al.*, 2024] as one of the baselines to highlight the improvements of our proposed adaptive rotations. Additionally, we include KVQuant [Hooper *et al.*, 2024] for PPL comparison, which leverages several techniques such as non-uniform quantization and per-vector dense-and-sparse quantization, demonstrating state-of-the-art PPL result. To ensure a fair comparison, for QuaRot we only quantize the KV cache, and for KVQuant, we adopt the scheme that preserves 0.5% FP16 outliers, which aligns the average bit-widths of our approach. As shown in Table 4, compared with the FP16 baseline, RotateKV shows a PPL degradation of 0.01 at 4-bit quantization, less than 0.1 at 3-bit, and less than 0.3 at 2-bit on LLaMA-2-13B. Compared to QuaRot, which exhibits significant PPL degradation at 2-bit quantization, our method remains effective even at extremely low bit widths. Compared to KVQuant, RotateKV consistently demonstrates a PPL improvement of around 0.1 at 2-bit quantization across all LLMs we tested. Notably, RotateKV uses the simpler integer quantization.

**GSM8K Evaluation**

The GSM8K dataset [Cobbe *et al.*, 2021] is widely used to evaluate the arithmetic reasoning capabilities of LLMs. This task presents significant challenges and evaluations on it can effectively show the impact of compression methods on model performance. As shown in the Table 5, RotateKV maintains strong CoT reasoning capabilities, with less than 1.7% performance degradation compared FP16 baseline. It outperforms existing methods even at lower average bit-widths, demonstrating the robustness of our approach. Notably, lower average bit-widths lead to higher compression ratio, enabling support for longer context lengths and larger batch sizes on the same GPU setup.

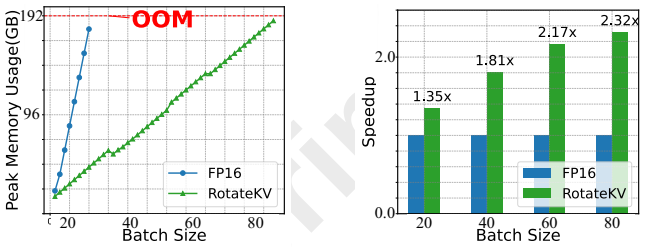
**LongBench and MileBench Evaluations**

To further demonstrate the robustness of RotateKV, we conduct experiments on eight long-context and multi-modal tasks selected from LongBench [Bai *et al.*, 2023b] and MileBench [Song *et al.*, 2024]. For comparison, we include KIVI [Liu

**Preprint – IJCAI 2025**: This is the accepted version made available for conference attendees.
**Do not cite**. The final version will appear in the IJCAI 2025 proceedings.

| Methods | Benchmarks | LongBench | | | | Benchmarks | MileBench | | | | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Tasks | QMSum | LCC | SAMSum | TriviaQA | Tasks | Scene Transition | Moving Attribute | Egocentric Navigation | Document VQA | |
| | Metrics | Rouge-L | Edit Sim | Rouge-L | F1 | Metrics | Accuracy | Accuracy | Accuracy | Accuracy | |
| FP16 | LLaMA-2 -7B | 21.1 | 66.7 | 41.3 | 87.4 | LLaVA-v1.5 -7B | 73.0 | 47.5 | 33.5 | 45.5 | 52.0 |
| KIVI | | 12.3 | 44.4 | 27.3 | 38.6 | | 34.5 | 25.5 | 21.0 | 19.0 | 27.8 |
| **RotateKV** | | **21.3** | **66.7** | **41.7** | **87.7** | | **67.5** | **37.0** | **27.0** | **29.0** | **47.2** |
| FP16 | LLaMA-2 -13B | 21.3 | 66.6 | 43.5 | 87.4 | LLaVA-v1.5 -13B | 70.5 | 50.0 | 26.5 | 46.0 | 51.5 |
| KIVI | | 20.6 | 45.6 | 33.1 | 75.7 | | 64.0 | 40.0 | 25.5 | 33.0 | 42.2 |
| **RotateKV** | | **21.4** | **66.6** | **43.5** | **87.4** | | **64.5** | **46.0** | **28.0** | **40.0** | **49.7** |
| FP16 | Mistral -7B | 20.4 | 67.3 | 43.2 | 89.2 | LLaVA-v1.6 -Mistral -7B | 71.0 | 44.0 | 34.5 | 38.0 | 51.0 |
| KIVI | | 19.0 | 58.3 | 41.0 | 81.6 | | **67.5** | **47.0** | 28.0 | 40.0 | 47.8 |
| **RotateKV** | | **20.5** | **67.4** | **42.9** | **89.2** | | 64.5 | 46.0 | **29.5** | **42.0** | **50.3** |

Table 6: Evaluations across a range of tasks from LongBench and MileBnech. The quantization is all set to 2-bit. For KIVI, the group size and the residual length are all set to 128. For RotateKV, the group size is set to 128.




(a) Memory evaluation. (b) Speed evaluation.

Figure 7: Efficiency analysis of RotateKV.

| LLaMA-2-13B | PPL ↓ |
|---|---|
| FP16 | 4.57 |
| Original Rotations | 6.96 (2.39 ↑) |
| + Pre-RoPE Grouped-Head Key Rotation | 5.67 (1.29 ↓) |
| + Attention-Sink-Aware Quantization | 5.52 (0.15 ↓) |
| + Outlier-Aware Rotations | 4.84 (0.68 ↓) |
| + Scale (FP8) & Zero-Point (INT8) | 4.84 |

Table 7: Ablation study of the proposed innovations. We evaluate the PPL of 2-bit quantization with a sequence length of 4096 using the LLaMA-2-13B model on the WikiText-2 dataset.

| Group Size | LLaMA-2-13B PPL ↓ | | | |
|---|---|---|---|---|
| | 16bit | 4bit | 3bit | 2bit |
| 128 | 4.57 | 4.58 | 4.64 | 4.84 |
| 64 | | 4.58 | 4.64 | 4.81 |
| 32 | | 4.58 | 4.66 | 4.75 |

Table 8: Experiments on quantization granularity using the LLaMA-2-13B on the WikiText-2 dataset with a sequence length of 4096.

*et al.*, 2024d], which uses per-channel Key quantization and shows negligible performance degradation with a group size of 32. As shown in Table 6, compared to the FP16 baseline, KIVI with a 128 quantization group size [Liu *et al.*, 2024d] exhibits significant performance degradation of 46.5% on the LLaMA-2-7B and LLaVA-v1.5-7B models. In contrast, our method demonstrates negligible average accuracy loss at the same quantization group size, with less than 1.4% performance loss on Mistral-7B and LLaVA-v1.6-Mistral-7B.

## 5.3 Ablation Study

**Ablation Study of the Proposed Innovations**

Starting with the original rotation method from QuaRot [Ashkboos *et al.*, 2024], we progressively incorporate the proposed innovations to evaluate their individual contributions. As shown in Table 7, the innovations introduced by RotateKV effectively mitigate PPL degradation.

**Ablation Study on Quantization Granularity**

To assess the impact of finer quantization granularity, we perform PPL evaluations with group sizes of 64 and 32. The results in Table 8 demonstrate that RotateKV continues to achieve better model performance preservation at smaller quantization group sizes. For instance, the PPL only drops by 0.18 on the LLaMA-2-13B model at 2-bit with a group size of 32.

## 5.4 Efficiency Analysis

In this section, we evaluate the efficiency of the current implementation. We use Triton [Tillet *et al.*, 2019] for the quantization and dequantization kernels, along with an optimized CUDA kernel for FWHT, following [Dao, 2024]. Evaluations are conducted on LLaMA-2-7B [Liu *et al.*, 2024a] with a 8-NVIDIA 4090D (24GB) setup, FlashAttention [Shah *et al.*, 2024] enabled. The batch size is progressively increased with an input length of 500 tokens until an out-of-memory (OOM) error occurs. As shown in Figure 7a, RotateKV reduces peak memory usage by 3.97× compared to the FP16 baseline and supports 5.75× larger batch sizes. In terms of speed, RotateKV achieves a 2.32× speedup during the decoding phase, as shown in Figure 7b. Notably, the efficiency can be further improved through techniques like kernel fusion.

# 6 Conclusion

In this paper, we explore the potential of rotation technique for 2-bit KV quantization. With the proposed innovations, RotateKV adaptively rotates the KV cache in an outlier-aware manner, demonstrating outstanding outliers control. Comprehensive evaluations show that RotateKV effectively preserves model performance even at high compression ratio, addressing the limitations of existing KV quantization methods and demonstrating state-of-the-art performance in both compression efficiency and accuracy. Future work will focus on optimize RotateKV's implementation to further reduce the overhead associated with online operation during LLM inference.

**Preprint – IJCAI 2025**: This is the accepted version made available for conference attendees.
**Do not cite**. The final version will appear in the IJCAI 2025 proceedings.

## References

[Ashkboos *et al.*, 2024] Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. *arXiv preprint arXiv:2404.00456*, 2024.

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

[Bai *et al.*, 2023b] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*, 2023.

[Chee *et al.*, 2024] Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. Quip: 2-bit quantization of large language models with guarantees. *Advances in Neural Information Processing Systems*, 36, 2024.

[Cobbe *et al.*, 2021] 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. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[Dao, 2024] Tri Dao. Fast hadamard transform in cuda, with a pytorch interface, 2024.

[Duanmu *et al.*, 2024] Haojie Duanmu, Zhihang Yuan, Xiuhong Li, Jiangfei Duan, Xingcheng Zhang, and Dahua Lin. Skvq: Sliding-window key and value cache quantization for large language models. *arXiv preprint arXiv:2405.06219*, 2024.

[Dubey *et al.*, 2024] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[Frantar *et al.*, 2022] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

[Fu *et al.*, 2024] Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. Data engineering for scaling language models to 128k context. *arXiv preprint arXiv:2402.10171*, 2024.

[Fu *et al.*, 2025] Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. Data engineering for scaling language models to 128k context. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2025.

[Guo *et al.*, 2024] Tianyu Guo, Druv Pai, Yu Bai, Jiantao Jiao, Michael I Jordan, and Song Mei. Active-dormant attention heads: Mechanistically demystifying extreme-token phenomena in llms. *arXiv preprint arXiv:2410.13835*, 2024.

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

[Hedayat and Wallis, 1978] A Hedayat and Walter Dennis Wallis. Hadamard matrices and their applications. *The annals of statistics*, pages 1184–1238, 1978.

[Hooper *et al.*, 2024] Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. Kvquant: Towards 10 million context length llm inference with kv cache quantization. *arXiv preprint arXiv:2401.18079*, 2024.

[Jiang *et al.*, 2023] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[Kang *et al.*, 2024] Hao Kang, Qingru Zhang, Souvik Kundu, Geonhwa Jeong, Zaoxing Liu, Tushar Krishna, and Tuo Zhao. Gear: An efficient kv cache compression recipe for near-lossless generative inference of llm, 2024.

[Lin *et al.*, 2024a] Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[Lin *et al.*, 2024b] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100, 2024.

[Lin *et al.*, 2024c] Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. Qserve: W4a8kv4 quantization and system co-design for efficient llm serving. *arXiv preprint arXiv:2405.04532*, 2024.

[Liu *et al.*, 2023] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023.

[Liu *et al.*, 2024a] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[Liu *et al.*, 2024b] Ruikang Liu, Haoli Bai, Haokun Lin, Yuening Li, Han Gao, Zhengzhuo Xu, Lu Hou, Jun Yao, and Chun Yuan. Intactkv: Improving large language model quantization by keeping pivot tokens intact. *arXiv preprint arXiv:2403.01241*, 2024.

[Liu *et al.*, 2024c] Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant–llm quantization with learned rotations. *arXiv preprint arXiv:2405.16406*, 2024.

**Preprint – IJCAI 2025**: This is the accepted version made available for conference attendees.
**Do not cite**. The final version will appear in the IJCAI 2025 proceedings.

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

[Merity *et al.*, 2016] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

[Saxena *et al.*, 2024] Utkarsh Saxena, Sayeh Sharify, Kaushik Roy, and Xin Wang. Resq: Mixed-precision quantization of large language models with low-rank residuals. *arXiv preprint arXiv:2412.14363*, 2024.

[Shah *et al.*, 2024] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *arXiv preprint arXiv:2407.08608*, 2024.

[Song *et al.*, 2024] Dingjie Song, Shunian Chen, Guiming Hardy Chen, Fei Yu, Xiang Wan, and Benyou Wang. Milebench: Benchmarking mllms in long context. *arXiv preprint arXiv:2404.18532*, 2024.

[Su *et al.*, 2024] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[Sun *et al.*, 2024] Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*, 2024.

[Tillet *et al.*, 2019] P. Tillet, H. T. Kung, and D. Cox. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pages 10–19. ACM, 2019.

[Touvron *et al.*, 2023] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[Vaswani, 2017] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[Xiao *et al.*, 2023a] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR, 2023.

[Xiao *et al.*, 2023b] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

[Yang *et al.*, 2024] June Yong Yang, Byeongwook Kim, Jeongin Bae, Beomseok Kwon, Gunho Park, Eunho Yang, Se Jung Kwon, and Dongsoo Lee. No token left behind: Reliable kv cache compression via importance-aware mixed precision quantization. *arXiv preprint arXiv:2402.18096*, 2024.

[Yu *et al.*, 2024] Zhongzhi Yu, Zheng Wang, Yonggan Fu, Huihong Shi, Khalid Shaikh, and Yingyan Celine Lin. Unveiling and harnessing hidden attention sinks: Enhancing large language models without training through attention calibration. *arXiv preprint arXiv:2406.15765*, 2024.

[Zhao *et al.*, 2023] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.