SignRot: LLM Quantization with Massive Outlier-Aware Sign-Adjusted Rotation

Jaehun Gim
Hanyang University
wogns3623@hanyang.ac.kr

Jaewoo Kim
Hanyang University
shrimp0426@hanyang.ac.kr

Gyuwan Kim
University of California, Santa Barbara
gyuwankim@ucsb.edu

Seo Yeon Park*
Hanyang University
seoyeonpark@hanyang.ac.kr

Abstract

While quantization compresses LLMs and accelerates inference, outlier activations impede efficient low-bit representation. In particular, massive outliers cause substantial performance degradation. Although rotation-based methods that rely on randomized Hadamard transforms for online inference aim to mitigate this, they can produce fragmented distributions with empty bins when a massive outlier dominates a token's activation. This fragmentation severely undermines uniform quantization performance. Moreover, massive outliers concentrate in the down-projection input, where the preceding nonlinearity requires the rotation to be applied at inference time. This constraint prevents training-based approaches such as SpinQuant from learning the online rotation at this position, as a learned dense matrix cannot use the Fast Walsh–Hadamard Transform and would require a full matrix multiplication during inference. To address this, we propose SignRot (Sign-adjusted Rotation), which identifies massive outlier channels and constructs a sign adjustment vector that unifies the signs of the corresponding Hadamard row, collapsing the bimodal distribution into a unimodal profile amenable to quantization. Since the aligned outlier shifts the activation mean away from zero, we pair SignRot with asymmetric activation quantization, whose zero-point absorbs this shift. As a lightweight, drop-in improvement compatible with any Hadamard-based quantization method, SignRot requires only a single calibration forward pass and adds only element-wise sign flips at inference, without an additional matrix multiplication. Applying SignRot to QuaRot achieves a WikiText2 perplexity of 5.88 and an average zero-shot reasoning accuracy of 72.73, surpassing SpinQuant on the LLaMA3-70B model. Code is available at https://github.com/HYU-ARK-Lab/SignRot.

1 Introduction

While large language models (LLMs) have achieved remarkable performance across diverse domains [32, 6, 25], their escalating scale presents substantial challenges in computational and memory requirements. Quantization has emerged as a promising efficiency technique due to its straightforward implementation and ability to maintain reliable performance under significant compression [17, 1, 19]. In particular, post-training quantization (PTQ) [11] has become a *de facto* standard, as it requires only a minimal calibration dataset without extensive retraining, making it highly practical for large-scale deployment. PTQ has been applied to quantize weights [11, 4], activations [38, 1], and the KV cache [12].

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

While PTQ methods are promising, the presence of large-magnitude outliers in activations, called massive outliers, remains a critical bottleneck, as they disproportionately distort the quantization process [38, 31, 1, 17]. To mitigate this issue, rotation-based PTQ methods, which transform the activation space to suppress outliers before quantization, have been explored [1, 19, 17]. These rotation matrices can generally be fused into adjacent weights offline, incurring no additional inference cost. However, for layers following nonlinear activations, such as FFN down-projection layers, the rotation cannot be fused into adjacent weights and thus typically requires online application during inference. To mitigate the resulting computational overhead, prior work has explored the use of *randomized* Hadamard matrices, which admit efficient implementation through the Fast Walsh–Hadamard Transform (FWHT) [1].

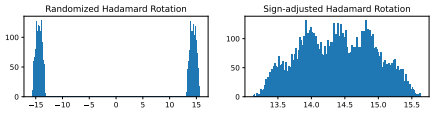


Figure 1: Distributions of token activations in LLaMA2-7B layer 2 down_proj containing massive outliers after rotation using *randomized* Hadamard and sign-adjusted Hadamard.

However, we find that when a massive outlier dominates a token's input activation, applying *randomized* Hadamard transforms produces a fragmented distribution with distinct positive and negative clusters, as shown in Figure 1, which directly degrades quantization performance. This is because the empty regions between these clusters inflate the activation quantization range and thus the step size. This is particularly critical at the FFN down-projection, where massive outliers are known to concentrate [30], and where existing rotation-based PTQ methods are constrained to use randomized Hadamard transforms for efficiency.

To address this, we propose a simple yet effective activation-aware sign adjustment to the Hadamard transform, named **Sign**-adjusted **Rot**ation (SignRot). By applying column-wise sign flips that unify the signs of the Hadamard row corresponding to the identified massive outlier channel, the fragmented distribution is restored to a bell-shaped profile amenable to quantization. Because the aligned outlier shifts the rotated activation's mean away from zero, SignRot pairs this adjustment with asymmetric activation quantization, whose zero-point absorbs the shift. SignRot is a lightweight, drop-in improvement compatible with any Hadamard-based quantization method, including QuaRot [1], SpinQuant [19], and DartQuant [29], and is applicable to both dense and MoE architectures. Since massive outlier channels are fixed across inputs [30], the sign adjustment vector needs only to be computed once via a single calibration forward pass; at inference, SignRot adds only an element-wise sign flip to the rotated activation, with no additional matrix multiplication. Furthermore, SignRot quantizes a 7B model in approximately 3 minutes with Round-to-Nearest (RTN), and around 15 minutes with GPTQ [11] on a single GPU, requiring no rotation training.

Our contribution is summarized as follows:

- We analytically characterize that randomized Hadamard transforms induce bimodal fragmentation in the presence of massive outliers, demonstrating that this fragmentation leads to quantization degradation at the FFN down-projection.
- We propose SignRot, a simple yet effective activation-aware sign adjustment to the Hadamard transform that directly addresses the fragmentation without requiring learned rotations. SignRot is a lightweight, drop-in improvement compatible with any Hadamard-based quantization method, requiring only a single calibration forward pass.
- SignRot improves quantization performance across diverse settings, including both dense and MoE architectures. On LLaMA3-70B with GPTQ, SignRot reduces the WikiText2 perplexity of QuaRot [1] from 6.16 to 5.88 under the same asymmetric activation quantization, and applied on top of SpinQuant [19], which relies on learned rotations, it further reduces perplexity from 6.38 to 5.45 and improves average zero-shot accuracy from 71.15 to 73.52.

2 Preliminaries and Related Work

Quantization formulations. Quantization is a technique that converts full-precision weights or activations (and optionally the KV cache) into low-precision (or discrete) values. This method is widely used to reduce the model size and inference latency. The general b-bit uniform quantization function $\mathcal{Q}_b$ maps an input matrix $\mathbf{X}$ into $\mathbf{X}_q = \mathcal{Q}_b(\mathbf{X}) = \Pi_{\Omega(b)}(\mathbf{X}/\Delta) + \mathbf{Z}$, where the subscript $_q$

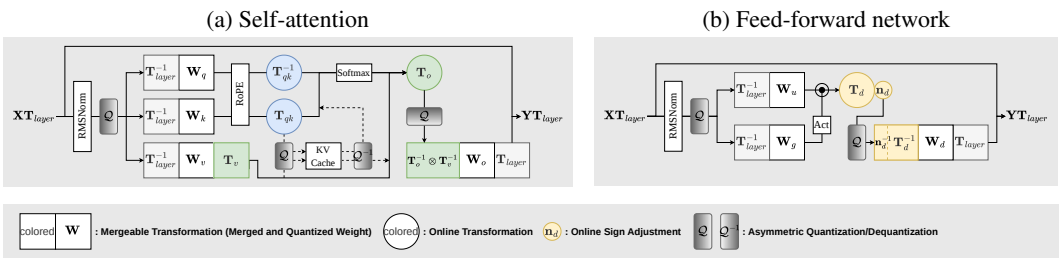


Figure 2: The overview of our proposed method SignRot in (a) self-attention and (b) feed-forward blocks, illustrated with a gated MLP (*e.g.,* LLaMA-style) as a representative architecture. Rotation matrices fall into two categories: mergeable transformation (*e.g.,* $\mathbf{T}_{layer}, \mathbf{T}_o \otimes \mathbf{T}_v$), which are fused into adjacent weights offline and incur no inference cost; and online transformation (*e.g.,* $\mathbf{T}_d$), which must be applied during inference due to preceding nonlinear operations. SignRot introduces a sign adjustment vector $\mathbf{n}_d$ at $\mathbf{T}_d$, the down-projection input where massive outliers concentrate, merging $\mathrm{diag}(\mathbf{n}_d)^{-1}$ into $\mathbf{W}_d$ offline while applying $\mathbf{n}_d$ online to the intermediate activation $\mathbf{X}_d$.

mean quantized, Δ is the step-size, $\mathbf{Z}$ is the zero point for asymmetric quantization, Π is a projection function onto the set of integers $\Omega(b) = \{0, 1, \ldots, 2^b - 1\}$ or $\Omega(b) = \{-2^{b-1}, \ldots, 2^{b-1} - 1\}$. For brevity, we omit b and denote $\mathcal{Q}(\cdot)$ as the general quantization function in the rest of this paper. The dequantized matrix $\widehat{\mathcal{Q}}(\mathbf{X}) = \mathbf{X}_{deq}$ is expressed as $\Delta \cdot (\mathbf{X}_q - \mathbf{Z})$, with scales and zero-points broadcast over their quantization groups. During inference, a linear layer $\mathbf{Y} = \mathbf{X}\mathbf{W}$ is approximated as $\mathbf{Y} \approx \mathbf{X}_{deq}\mathbf{W}_{deq}$. In this work, we primarily use per-token asymmetric quantization for activations and per-channel symmetric quantization for weights.

Channel outlier and scaling methods Previous studies [8, 38, 28] have focused on handling channel outliers that appear in specific feature channels across entire token sequences. These methods primarily leverage *scale invariance*, which utilizes the fact that a linear layer $\mathbf{Y} = \mathbf{X}\mathbf{W}$ remains mathematically equivalent when activations are scaled by a diagonal matrix $\mathbf{S}$ and weights are inversely scaled: $\mathbf{Y} = (\mathbf{X}\mathbf{S})(\mathbf{S}^{-1}\mathbf{W})$.

While LLM.int8 [8] initially managed these outliers through mixed-precision decomposition, its implementation proved hardware-unfriendly. To address this, SmoothQuant [38] introduced a technique to migrate the quantization difficulty of channel outliers from activations to weights via channel-wise scaling. This direction was further refined by OmniQuant [28], which improved quantization performance by learning optimal channel-wise scaling factors.

However, scaling methods that only adjust the per-channel magnitude cannot effectively handle a newly discovered type of outlier [30, 18] called the massive outlier. These outliers concentrate in specific tokens and channels, particularly in FFN layers, and are known to be significant in quantizing activations to 4 bits or less [17].

Massive outlier and rotation methods To mitigate the effects of massive outliers, recent studies [1, 19, 29] leverage *transformation invariance*. Unlike scaling, which only performs element-wise adjustment, rotation-based methods use an orthogonal transformation $\mathbf{R}$ to redistribute the outlier energy across all available channels: $\mathbf{Y} = (\mathbf{X}\mathbf{R})(\mathbf{R}^{-1}\mathbf{W})$. The applicability and optimization of these rotations are fundamentally governed by their position within the Transformer [35] block.

When a rotation follows a linear operation, it can be fused directly into adjacent weights during the PTQ phase, incurring no additional inference cost. This offline mergeability allows for intensive optimization through training to maximize accuracy. For instance, SpinQuant [19] optimizes the rotation matrix using CayleySGD [15]. DartQuant [29] further improves the efficiency of rotation matrix optimization by constraining the post-rotation activation distribution, which reduces both computational cost and overfitting risk.

In contrast, rotations that follow nonlinear operations, such as the FFN down-projection input (*i.e.,* T_d in Figure 2, where a LLaMA-style architecture is used as a representative example), cannot be fused and must be applied during inference. To maintain inference efficiency at these positions, existing methods are constrained to use the Randomized Hadamard Transform (RHT) to leverage the Fast Walsh–Hadamard Transform (FWHT) [10]. Various methodologies have been employed to

implement this framework. QuaRot [1] serves as a foundation for training-free rotation, effectively removing outliers using RHT, while DFRot [37] refines the rotation using a small calibration dataset to minimize quantization errors in a near-closed-form manner.

While these training-free methods offer fast setup using a minimal calibration step, they remain agnostic to the specific polarity of massive outliers. As we will analyze in Section 3, this disregard for outlier polarity induces a systematic bimodal fragmentation when using RHT at the FFN down-projection. This serves as a critical source of quantization degradation that cannot be resolved without considering the signs of the outliers.

3 Analysis of Bimodal Fragmentation in RHT under Massive Outliers

The *randomized* Hadamard transform (RHT) combines independent random signs with the uniform structure of the Hadamard matrix. In the absence of dominant outliers, its output can have a bell-shaped distribution favorable for quantization [34]. The Gaussian approximation used here requires the conditions discussed below. The Gaussian approximation under RHT relies on the Central Limit Theorem (CLT). Independent random signs yield zero-mean summands. In this setting, the Lindeberg condition requires that no single summand contribute a non-negligible fraction of the total variance in the limit [2]. This assumption can fail when one channel retains a non-vanishing fraction of the activation energy; that channel then contributes an independent two-valued term, so the limit is not Gaussian.

Formally, let $\mathbf{x}_t \in \mathbb{R}^{C_{in}}$ be the activation vector for token t, with signed outlier value $M_t = x_{t,c_{max}}$ at channel c_{max}. When the RHT is applied, the output is computed as $\mathbf{y}_t = \mathbf{x}_t \cdot \mathrm{diag}(\mathbf{r}) \cdot \mathbf{H}$, where $\mathbf{H}$ is an orthonormalized Hadamard matrix and $r_i \in \{\pm 1\}$ are independent random signs. The j-th element of the transformed vector decomposes as follows:

$$y_{t,j} = M_t \cdot r_{c_{max}} h_{c_{max},j} + \sum_{i \neq c_{max}} x_{t,i} r_i h_{i,j} \tag{1}$$

where $h_{i,j}$ is the (i,j)-th entry of the Hadamard matrix $\mathbf{H}$ and $x_{t,i}$ is the i-th element of $\mathbf{x}_t$. For a fixed output channel j, the residual has variance $\sigma^2_{rest} = C_{in}^{-1} \sum_{i \neq c_{max}} x^2_{t,i}$ over the random signs. When the residual satisfies its own Lindeberg condition, we approximate it by $\mathcal{N}(0, \sigma^2_{rest})$. The independent outlier term takes values $\pm M_t/\sqrt{C_{in}}$ with equal probability, giving the Gaussian-mixture approximation

$$y_{t,j} \overset{\text{approx.}}{\sim} \frac{1}{2}\mathcal{N}\left(+\frac{M_t}{\sqrt{C_{in}}}, \sigma^2_{rest}\right) + \frac{1}{2}\mathcal{N}\left(-\frac{M_t}{\sqrt{C_{in}}}, \sigma^2_{rest}\right). \tag{2}$$

This mixture describes a distribution over random signs, not an exact law for the channels of one fixed transform. For a balanced outlier row, half the channels receive each shift; interpreting their histogram as this mixture additionally assumes approximately Gaussian residuals within both groups (Appendix A). For the full sum, the Lindeberg condition reduces to $\max_i x^2_{t,i} / \sum_i x^2_{t,i} \to 0$. Its violation is asymptotic; the finite threshold below is exact only within the Gaussian-mixture model. An equally weighted mixture of two Gaussians with a common variance σ^2 is bimodal if and only if their means are more than 2σ apart [27]. Applying this to Equation 2, *i.e.,* $|M_t|/\sqrt{C_{in}} > \sigma_{rest}$, yields

$$M_t^2 > \sum_{i \neq c_{max}} x^2_{t,i}, \tag{3}$$

that is, the outlier channel must carry more energy than all remaining channels combined. Since this requires $M_t^2 > \|\mathbf{x}_t\|_2^2/2$, at most one channel per token can satisfy Equation 3. We refer to such a channel as the dominant massive outlier channel of the token; it is the channel that SignRot targets. Accordingly, a massive outlier token is defined as a token whose largest channel satisfies Equation 3 and reaches at least half of the peak activation magnitude of the layer, provided that this peak is at least 100. This energy-based criterion differs from the magnitude-based criterion of Sun et al. [30]; Appendix C compares the two criteria.

For the massive outlier token of LLaMA2-7B in Table 7 ($C_{in} = 11{,}008$), the threshold is $\sqrt{C_{in}}\,\sigma_{rest} \approx 58$, whereas the observed outlier is $M_t = 1510$, about $26\times$ larger, consistent with the clearly separated clusters in Figure 1. The full derivation is provided in Appendix A. Since the

threshold is derived under the two-Gaussian approximation, energy dominance should be read as an indicator of fragmentation rather than an exact test for empirical bimodality; in particular, several large residual channels can invalidate the two-component model. As illustrated in Figure 1, bimodal fragmentation can degrade quantization performance by widening the activation range. For an input activation token $\mathbf{x}_t$ quantized with a shared step size Δ_t, nearest rounding without clipping gives $|\widehat{\mathcal{Q}}(x_{t,j}) - x_{t,j}| \leq \Delta_t/2$ for each channel. The mean squared error (MSE) of the quantized input is therefore

$$\mathrm{MSE}_t := \frac{1}{C_{in}} \left\| \widehat{\mathcal{Q}}(\mathbf{x}_t) - \mathbf{x}_t \right\|_2^2 = \frac{1}{C_{in}} \sum_{j=1}^{C_{in}} \left(\widehat{\mathcal{Q}}(x_{t,j}) - x_{t,j} \right)^2 \leq \frac{\Delta_t^2}{4}. \tag{4}$$

Since the two component means are separated by $2|M_t|/\sqrt{C_{in}}$, the per-token range grows by at least this amount in addition to the spread of the residual, which increases Δ_t at a fixed bit width and hence the rounding error in Equation 4. The gap between the clusters additionally leaves the central quantization bins unused, while the realized MSE also depends on grid alignment and clipping. Although massive outlier tokens are rare, the fragmentation they cause substantially increases the quantization error of the layer in which they first appear (Appendix C). The range inflation itself, however, follows directly from the sign pattern of the outlier row, and removing it is the goal of the sign adjustment introduced next. We provide activation distribution statistics for massive outlier tokens and non-outlier tokens in Appendix B.

4 Proposed Method: SignRot

The bimodal fragmentation analyzed above arises from the alternating signs of the Hadamard entries $h_{c_{max},j} \in \{\pm 1/\sqrt{C_{in}}\}$ corresponding to the outlier channel c_{max}. We therefore propose **Sign**-adjusted **Rot**ation (SignRot), which unifies the signs of the c_{max}-th row of $\mathbf{H}$, such that the selected outlier contribution becomes a constant shift $M_t/\sqrt{C_{in}}$ across all output channels. If the adjusted residual is approximately Gaussian, the resulting distribution is approximately unimodal.

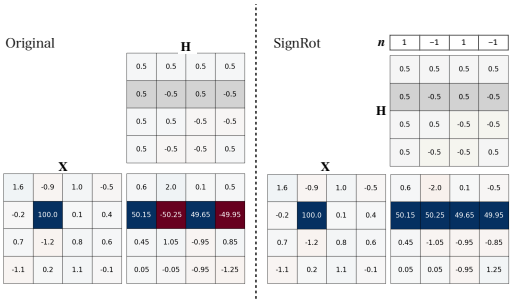


Figure 3: Effect of sign adjustment on the *randomized* Hadamard matrix under a massive outlier. Without adjustment (left), large outlier activation produces fragmented outputs of mixed sign. With sign alignment via $\mathrm{diag}(\boldsymbol{n})$ (right), the outlier contribution becomes a uniform positive shift, collapsing the bimodal distribution.

4.1 Identifying the massive outlier channel

To identify the target channel, we examine the activation matrix $\mathbf{X} \in \mathbb{R}^{T \times C_{in}}$ over a calibration set and locate the channel with the largest absolute activation:

$$c_{\max} = \arg\max_c \left(\max_t |x_{t,c}| \right). \tag{5}$$

We identify one channel in a calibration forward pass and reuse it for subsequent inputs. This choice follows the observed stability of massive outlier channels [30]: in the first massive outlier layer of every model we examine, all massive outlier tokens share the same dominant massive outlier channel, which holds 87–99.9% of the token energy (Table 11, Appendix C). When a token carries further massive outlier channels, *i.e.,* channels that each carry more energy than all smaller channels combined (Equation 3 applied recursively), SignRot still removes the split induced by the selected channel (Appendix D). Aligning this single channel therefore removes most of the fragmentation, although some later layers do not share the same dominant massive outlier channel (Appendix H).

4.2 Sign-adjusted Hadamard transformation

Given c_{max}, we define the sign adjustment vector $\mathbf{n} \in \{+1, -1\}^{C_{in}}$ as follows:

$$n_j := \mathrm{sign}(h_{c_{max},j}) = \begin{cases} 1 & \text{if } h_{c_{\max},j} \geq 0, \\ -1 & \text{otherwise,} \end{cases} \tag{6}$$

We then define the sign-adjusted Hadamard transformation matrix $\mathbf{H}_{\mathrm{adj}} \in \mathbb{R}^{C_{in} \times C_{in}}$ as follows:

$$\mathbf{H}_{\mathrm{adj}} = \mathbf{H}\ \mathrm{diag}(\mathbf{n}), \tag{7}$$

where right-multiplication by $\mathrm{diag}(\mathbf{n})$ flips the sign of each column j by n_j, so that every entry of the $c_{\max}$-th row of $\mathbf{H}_{\mathrm{adj}}$ becomes $+1/\sqrt{C_{in}}$. We demonstrate this effect in Figure 3. Without sign adjustment, the outlier row (*i.e.,* containing the massive activation ≈ 100.0) produces mixed-sign outputs that directly reflect the alternating signs of the corresponding Hadamard row (Figure 3 left). After applying $\mathrm{diag}(\mathbf{n})$, the same row produces uniformly positive outputs (Figure 3 right). Consequently, the selected outlier term $M_t h_{c_{max},j} n_j = M_t/\sqrt{C_{in}}$ becomes a constant shift for all j, removing this channel's sign-induced split. In practice, we apply this adjustment at the down-projection input ($\mathbf{T}_d$ in Figure 2), where massive outliers concentrate [30]. We denote the corresponding vector as $\mathbf{n}_d$.

Quantization with sign-adjusted Hadamard transformation. Since $\mathrm{diag}(\mathbf{n})$ is an involutory matrix (*i.e.,* $\mathrm{diag}(\mathbf{n})^{-1} = \mathrm{diag}(\mathbf{n})$), the inverse is given by $\mathbf{H}_{\mathrm{adj}}^{-1} = \mathrm{diag}(\mathbf{n})\,\mathbf{H}^{-1}$. This allows the adjusted transformation to be inserted into a linear layer without altering the computation since $\mathbf{Y} = \mathbf{X}\mathbf{W} = (\mathbf{X}\mathbf{H}_{\mathrm{adj}})(\mathbf{H}_{\mathrm{adj}}^{-1}\mathbf{W})$ holds. For clarity, the random-sign factor $\mathrm{diag}(\mathbf{r})$ is omitted here. Quantization is applied to each factor independently; $\mathbf{Y}_q$ denotes the resulting integer accumulation:

$$\mathbf{Y}_q = \mathcal{Q}(\mathbf{X}\mathbf{H}_{\mathrm{adj}})\,\mathcal{Q}\Big(\mathbf{H}_{\mathrm{adj}}^{-1}\mathbf{W}\Big)\,. \tag{8}$$

Asymmetric–Symmetric matrix multiplication SignRot pairs sign adjustment with asymmetric activation quantization to represent the resulting shifted distribution. Sign adjustment changes the signs of the Hadamard outputs while preserving their absolute values; it therefore leaves the max-absolute scale of symmetric quantization unchanged. Asymmetric quantization uses a zero-point to accommodate the shift and can allocate quantization levels to the shifted range. Asymmetric activation quantization is therefore an integral part of SignRot rather than an optional refinement. To maintain the efficiency of activation quantization, we use Asymmetric–Symmetric matrix multiplication (Asym–Symm matmul, where activations are quantized asymmetrically and weights symmetrically) defined as follows:

$$\mathbf{Y}_{deq} = \Delta_{\mathbf{X}}(\mathbf{X}_q\mathbf{W}_q - \mathbf{Z}_{\mathbf{X}}\mathbf{W}_q)\Delta_{\mathbf{W}} \tag{9}$$

where $\mathbf{Y}_{deq}$ denotes the dequantized output, $\Delta_{\mathbf{X}}$ and $\Delta_{\mathbf{W}}$ are diagonal matrices of per-token activation scales and per-output-channel weight scales, respectively, and $\mathbf{Z}_{\mathbf{X}}$ contains the activation zero-points. The first term in Equation 9 is an integer dot product identical to Symmetric–Symmetric matrix multiplication (Symm–Symm matmul, where both activations and weights are quantized symmetrically). For per-token activation quantization, $\mathbf{Z}_{\mathbf{X}} = \mathbf{z}_{\mathbf{X}}\mathbf{1}^\top$, where $\mathbf{z}_{\mathbf{X}}$ is a column vector of token zero-points and $\mathbf{1} \in \mathbb{R}^{C_{in}}$ is the all-ones vector. The second term therefore factorizes as $\mathbf{z}_{\mathbf{X}}(\mathbf{1}^\top\mathbf{W}_q)$. The weight column sum $\mathbf{1}^\top\mathbf{W}_q$, with $\mathbf{W}_q \in \mathbb{Z}^{C_{in}\times C_{out}}$, is input-independent and precomputed offline. During inference, Asym–Symm matmul computes the rank-1 outer product with $\mathbf{z}_{\mathbf{X}}$ and subtracts it from the integer matrix product. This correction costs $O(T\,C_{out})$, compared with $O(T\,C_{in}C_{out})$ for the matrix multiplication. Asymmetric quantization also estimates a per-token zero-point alongside the scale required by symmetric quantization.

5 Experiments

5.1 Experimental setup

Models and datasets We apply SignRot on the LLaMA2 (7B, 13B, 70B) [33] and LLaMA3 (8B, 70B) [9] model families, and additionally on Mistral-7B-v0.3 [13] and Qwen3-30B-A3B [39]. To assess our method, we evaluate quantized models in language modeling and reasoning tasks. We report perplexity on WikiText2 [20] and C4 [24] datasets to evaluate language modeling. Furthermore, we measure the accuracy on six zero-shot reasoning tasks: ARC (Easy and Challenge) [5], HellaSwag [41], LAMBADA(OpenAI) [23], PIQA [3], and WinoGrande [26].

Baselines We evaluate SignRot by applying it on top of three rotation-based methods that employ online Hadamard transforms: QuaRot [1], SpinQuant [19], and DartQuant [29]. QuaRot applies randomized Hadamard transforms without any training, while SpinQuant and DartQuant further optimize the rotation matrices to minimize quantization error.

Table 1: Comparison of the perplexity score on WikiText2 and C4 (**PPL**), and averaged accuracy on six zero-shot reasoning tasks (**Avg. 0-shot**) for LLaMA families. '+SignRot' indicates that our method has been applied. Full results are in the Appendix F. Note that QuaRot uses symmetric activation quantization.

Method	W Quantizer	2-7B			2-13B			2-70B			3-8B			3-70B		
		Wiki2 PPL↓	C4 PPL↓	Avg. 0-shot↑	Wiki2 PPL↓	C4 PPL↓	Avg. 0-shot↑	Wiki2 PPL↓	C4 PPL↓	Avg. 0-shot↑	Wiki2 PPL↓	C4 PPL↓	Avg. 0-shot↑	Wiki2 PPL↓	C4 PPL↓	Avg. 0-shot↑
FP16	-	5.47	7.26	69.79	4.88	6.73	72.55	3.32	5.71	77.05	6.14	9.45	73.23	2.86	7.17	79.95
QuaRot	RTN	8.56	11.86	57.73	6.10	8.67	66.25	4.14	6.42	73.47	10.60	17.19	61.34	55.44	79.48	35.36
QuaRot+SignRot	RTN	7.93	11.21	60.43	6.00	8.48	67.51	4.05	6.32	73.86	9.68	15.72	64.17	41.47	58.26	38.52
QuaRot	GPTQ	6.10	8.32	65.01	5.40	7.54	68.91	3.79	6.12	75.68	8.16	13.38	65.79	6.60	12.87	70.45
QuaRot+SignRot	GPTQ	6.09	8.31	67.19	5.32	7.43	70.70	3.74	6.04	75.68	7.85	12.74	67.27	5.88	11.24	72.73
SpinQuant	GPTQ	6.08	8.25	66.27	5.39	7.46	69.94	3.73	6.04	75.65	7.85	12.71	67.96	6.38	13.98	71.15
SpinQuant+SignRot	GPTQ	6.09	8.27	67.07	5.37	7.45	70.84	3.72	6.04	75.79	7.87	12.60	69.36	5.45	10.54	73.52
DartQuant	GPTQ	5.97	8.07	66.98	5.32	7.39	70.41	3.66	6.01	75.61	7.49	12.16	69.27	4.93	9.45	76.39
DartQuant+SignRot	GPTQ	5.94	8.04	67.27	5.31	7.37	70.59	3.66	6.00	75.86	7.40	12.05	69.03	4.86	9.26	76.02

Implementation details We implemented SignRot using HuggingFace Transformers [36] and PyTorch [22]. On a single A100 GPU, SignRot only adds 5 minutes of quantization time for the LLaMA2-70B model, which is negligible compared to the total quantization time. The peak memory usage during quantize LLaMA2-70B is max 7 GB for the rotation process and max 40 GB for the GPTQ stage. We use CUTLASS [21] for 4-bit matrix multiplication and FlashInfer [40] for KV cache quantization to evaluate the inference performance of our fully quantized model. Total quantization time is measured on a single NVIDIA A100 GPU, while inference speedup and memory usage are measured on an NVIDIA RTX 3090 GPU.

Quantization configurations We report results using both Round-to-Nearest (RTN) and GPTQ [11] for weight quantization; our GPTQ implementation uses 128 calibration samples randomly selected from the WikiText-2 dataset. To address the memory-intensive nature of self-attention, we apply group-wise asymmetric quantization to the KV cache with a group size of the head dimension, following [1]. For each baseline, +SignRot uses the same quantization configuration as the baseline, except that +SignRot applies per-token asymmetric activation quantization and its clipping ratios are selected by hyperparameter tuning. QuaRot uses per-token symmetric activation quantization following its official setting [1], whereas SpinQuant and DartQuant already use asymmetric activation quantization. Appendix E gives the detailed quantization configuration of each baseline and of SignRot. Sign adjustment vectors are computed using the first sample of the GPTQ calibration dataset during the GPTQ calibration phase. For RTN, we compute the sign adjustment vector using a single sample randomly extracted from the WikiText-2 dataset. We confirm that the choice of calibration data has minimal impact on performance in Table 5.

5.2 Main results

Results on LLaMA families. We report the main results in Table 1, where SignRot is applied on top of three rotation-based baselines, respectively, on LLaMA2 (7B/13B/70B) and LLaMA3 (8B/70B) under both RTN and GPTQ weight quantization. We evaluate language modeling perplexity on WikiText2 and C4, and zero-shot reasoning task accuracy.

For language modeling tasks, QuaRot+SignRot generally reduces perplexity over the QuaRot baseline. The improvement is especially pronounced for LLaMA3-70B, where Wiki2 perplexity drops from 6.60 to 5.88 under GPTQ (from 6.16 to 5.88 under the same asymmetric activation quantization; Table 4), surpassing SpinQuant [19] (6.38). For LLaMA2-13B with GPTQ, QuaRot+SignRot achieves a Wiki2 PPL of 5.32, matching DartQuant [29] and outperforming SpinQuant (5.39). On LLaMA3-8B, QuaRot+SignRot +GPTQ reduces PPL from 8.16 to 7.85, matching SpinQuant and narrowing the gap to DartQuant (7.49). Applying SignRot on top of SpinQuant and DartQuant also yields perplexity reductions in most settings, demonstrating that sign adjustment is broadly complementary to other rotation-based methods.

For zero-shot reasoning tasks, we make the following observations: under RTN quantization, QuaRot+SignRot generally outperforms QuaRot, *i.e.,* achieving an average accuracy improvement of up to 2.83 points on LLaMA3-8B. On LLaMA2, this gain decreases with model size (2.70, 1.26, and 0.39 points for 7B, 13B, and 70B), consistent with the greater robustness of larger models to quantization [16, 14]: the accuracy drop from FP16 to QuaRot under RTN decreases from 12.06 to 6.30 and

Table 2: Zero-shot reasoning task accuracy (**0-shot**) and perplexity scores on WikiText2 and C4 (**PPL**) for 4-bit weight, activation, and KV-cache quantized Mistral and Qwen3. Note that QuaRot uses symmetric activation quantization.

Model	Method	W Quantizer	Wiki2 PPL↓	C4 PPL↓	ARC-C 0-shot↑	ARC-E 0-shot↑	HellaSwag 0-shot↑	LAMBADA 0-shot↑	PIQA 0-shot↑	Winogrande 0-shot↑	Avg. 0-shot↑
Mistral-7B-v0.3	FP16	-	5.32	8.47	52.30	78.20	80.41	75.32	82.26	73.88	73.73
	QuaRot	GPTQ	5.80	9.19	48.89	76.13	78.09	73.18	79.86	72.37	71.42
	QuaRot+SignRot	GPTQ	5.71	9.09	48.72	76.09	78.95	72.64	80.30	72.45	71.53
Qwen3-30B-A3B	FP16	-	8.70	14.05	56.57	79.25	77.63	64.76	80.47	69.38	71.34
	QuaRot	RTN	11.70	18.21	47.18	68.90	69.00	52.73	75.57	63.38	62.79
	QuaRot+SignRot	RTN	11.21	17.63	48.04	69.70	70.04	53.97	75.30	62.04	63.18
	QuaRot	GPTQ	10.16	16.22	48.89	73.82	71.51	57.25	75.68	63.54	65.11
	QuaRot+SignRot	GPTQ	9.79	15.93	47.70	72.10	72.33	60.99	76.99	64.72	65.80

Table 3: End-to-end inference latency and peak memory of LLaMA2-7B on a single RTX 3090 GPU at batch size 1: prefill of a 2048-token prompt followed by 128 decoding steps. Both quantized models use per-token asymmetric activation quantization.

Method	Prefill		Decode		E2E		Peak memory	
	ms	Speedup	ms/token	Speedup	ms	Speedup	GB	Saving
FP16	496.7	1.00×	24.01	1.00×	3626.6	1.00×	13.84	1.00×
QuaRot	283.4	1.75×	14.92	1.61×	2194.0	1.65×	4.33	3.19×
QuaRot+SignRot	287.3	1.73×	14.97	1.60×	2203.5	1.65×	4.33	3.19×
Overhead of SignRot	+1.37%		+0.34%		+0.43%		+0.02%	

3.58 points, leaving less quantization error for SignRot to recover. Under GPTQ, QuaRot+SignRot surpasses SpinQuant [19] by 0.92 points on LLaMA2-7B (67.19 vs. 66.27) and by 0.76 points on LLaMA2-13B (70.70 vs. 69.94), even though SpinQuant relies on learned rotation matrices. On the larger LLaMA3-70B model, QuaRot+SignRot+GPTQ improves average accuracy from 70.45 to 72.73 (from 72.28 to 72.73 under the same asymmetric activation quantization; Table 4), again exceeding SpinQuant (71.15). Furthermore, applying SignRot to SpinQuant and DartQuant [29] baselines yields additional gains in most settings, demonstrating the broad applicability of our approach.

Results on other models To validate the generalizability of our method, we apply SignRot to two additional models: Mistral-7B-v0.3 [13] and Qwen3-30B-A3B [39], and show results in Table 2. SignRot generally improves aggregate performance over the QuaRot baseline: it lowers perplexity in all settings and raises the average zero-shot accuracy, although some individual tasks decrease (*e.g.,* LAMBADA on Mistral-7B-v0.3 and ARC on Qwen3-30B-A3B with GPTQ). On Mistral-7B-v0.3, SignRot with GPTQ reduces the WikiText2 perplexity from 5.80 to 5.71 while keeping a similar average accuracy (71.53 vs. 71.42). On Qwen3-30B-A3B, SignRot with GPTQ outperforms QuaRot with GPTQ by 0.69 points in average accuracy, suggesting that our method generalizes across model families (Mistral) and scales to larger MoE architectures (Qwen3-30B-A3B).

5.3 Efficiency

End-to-end inference We measure the end-to-end inference latency of the full LLaMA2-7B model on a single NVIDIA RTX 3090 GPU with our deployment kernels, prefilling a 2048-token prompt and then decoding 128 tokens at batch size 1 (Table 3). Recent rotation-based methods such as SpinQuant [19] and DartQuant [29] adopt per-token asymmetric activation quantization, and SpinQuant+SignRot and DartQuant+SignRot in Table 1 use the same asymmetric quantizer as their baselines. We therefore measure the cost of SignRot against QuaRot with the same asymmetric quantizer. SignRot adds only 1.37% to the prefill latency, 0.34% to the per-token decoding latency, and 0.43% to the end-to-end latency, and increases the peak memory by only 0.02%. Over FP16, QuaRot+SignRot achieves 1.73×, 1.60×, and 1.65× speedups in prefill, decoding, and end-to-end latency, respectively, with a 3.19× memory saving.

These results confirm that adding SignRot does not introduce significant computational overhead, allowing us to maintain the substantial speedup and memory savings benefits of quantization while achieving improved accuracy on language modeling and reasoning tasks. Results with symmetric

Table 4: Ablation study on the impact of asymmetric activation quantization with GPTQ. LLaMA2-7B and LLaMA3-8B values are averaged over ten seeds; LLaMA3-70B is a single run, and its symmetric QuaRot row is taken from Table 1.

Model	Method	Act asym	PPL (↓)		Acc (↑)						
			Wiki2	C4	ARC-C	ARC-E	HellaSwag	LAMBADA	PIQA	Winogrande	Avg.
2-7B	QuaRot	✗	6.12	8.31	42.41	70.26	72.62	70.61	76.99	66.35	66.54
	QuaRot+SignRot	✗	6.12	8.32	41.73	69.48	72.37	70.50	77.15	66.74	66.33
	QuaRot	✓	6.06	8.23	42.42	70.16	72.70	70.93	77.34	66.13	66.62
	QuaRot+SignRot	✓	6.05	8.23	42.08	69.89	72.70	71.13	77.20	66.91	66.65
3-8B	QuaRot	✗	8.15	13.28	44.71	70.62	73.66	68.18	75.56	67.29	66.67
	QuaRot+SignRot	✗	8.09	13.15	45.12	70.67	73.74	67.47	75.71	67.02	66.62
	QuaRot	✓	7.94	12.92	44.99	71.30	74.00	69.08	76.03	67.55	67.16
	QuaRot+SignRot	✓	7.88	12.81	45.73	72.13	74.25	68.62	76.75	68.18	67.61
3-70B	QuaRot	✗	6.60	12.87	49.49	74.37	77.22	71.69	78.89	71.03	70.45
	QuaRot	✓	6.16	11.96	50.94	76.47	78.66	72.04	81.07	74.51	72.28
	QuaRot+SignRot	✓	5.88	11.24	52.65	77.36	79.48	72.09	80.96	73.88	72.73

activation quantization and the overhead of SignRot on the down-projection layer alone are provided in Appendix G.

5.4 Ablation studies

Symmetric activation quantization To isolate the contribution of asymmetric activation quantization to our method, we compare symmetric and asymmetric per-token quantization for activations on LLaMA2-7B and LLaMA3-8B with GPTQ, averaged over ten seeds, with results shown in Table 4. Under symmetric activation quantization, the effect of SignRot varies across models. Switching to asymmetric per-token activation quantization yields an additional gain for QuaRot+SignRot on both models: average zero-shot accuracy improves by 0.99 points on LLaMA3-8B (66.62 → 67.61) and by 0.32 points on LLaMA2-7B (66.33 → 66.65), while WikiText2 perplexity drops from 8.09 to 7.88 and from 6.12 to 6.05, respectively. On LLaMA3-70B (single run, Table 4), switching QuaRot from symmetric to asymmetric activation quantization changes WikiText2 perplexity, C4 perplexity, and average zero-shot accuracy from 6.60, 12.87, and 70.45 to 6.16, 11.96, and 72.28. Adding SignRot under the same asymmetric quantization gives 5.88, 11.24, and 72.73. These results confirm that asymmetric activation quantization captures the inherent asymmetry in post-rotation activation distributions and is a key driver of the overall performance improvements reported in our main results.

Insensitivity to calibration data To evaluate the sensitivity of our method to the choice of calibration data, we compare the performance of QuaRot+SignRot when using different datasets to compute the sign adjustment vector $\mathbf{n}$. Specifically, we use WikiText-2 and C4 as calibration datasets, each with three different random seeds to select calibration samples. As shown in Table 5, across the two datasets and three seeds, the WikiText2 and C4 perplexities vary by at most 0.06 and 0.08, and the average accuracy by 0.9 points, indicating that SignRot is insensitive to the choice of calibration data.

Table 5: Performance of QuaRot+SignRot on LLaMA3-8B with GPTQ when the sign adjustment vector is computed from different calibration datasets and seeds.

Calibration Dataset	Wiki2 PPL	C4 PPL	Avg. Acc
Wiki2 (Seed 0)	7.85	12.74	66.66
Wiki2 (Seed 1)	7.87	12.80	67.04
Wiki2 (Seed 2)	7.84	12.77	67.41
C4 (Seed 0)	7.88	12.78	67.26
C4 (Seed 1)	7.82	12.77	66.52
C4 (Seed 2)	7.87	12.82	67.25

Adjustment on other linear layers We evaluated sign adjustment at different projection inputs of LLaMA3-8B under GPTQ with per-token asymmetric activations, averaged over six seeds (Table 6). We adjusted the inputs of the q/k/v, out, up/gate, and down projections individually. We also tested combinations within the self-attention block ($\tilde{\mathbf{X}}_{qkv} + \tilde{\mathbf{X}}_o$) and the MLP block ($\tilde{\mathbf{X}}_{ug} + \tilde{\mathbf{X}}_d$), the inputs to the out and down projections together ($\tilde{\mathbf{X}}_o + \tilde{\mathbf{X}}_d$), and all inputs together. Among single positions, adjusting $\tilde{\mathbf{X}}_d$ gave the lowest perplexity (7.898 Wiki2, 12.869 C4) and the highest accuracy (67.74), whereas adjusting only $\tilde{\mathbf{X}}_{qkv}$, $\tilde{\mathbf{X}}_{ug}$, or $\tilde{\mathbf{X}}_o$ gave 7.957–7.974 Wiki2 PPL and 66.94–67.21 accuracy. Every configuration that included $\tilde{\mathbf{X}}_d$ reached a similar perplexity (7.876–7.898), and adjusting all inputs gave the lowest perplexity but no additional accuracy (67.49 vs. 67.74). We therefore apply sign

Table 6: Sign adjustment applied at different projection inputs of LLaMA3-8B with GPTQ weight quantization and per-token asymmetric activation quantization. $\tilde{\mathbf{X}}_{qkv}$, $\tilde{\mathbf{X}}_{o}$, $\tilde{\mathbf{X}}_{ug}$, and $\tilde{\mathbf{X}}_{d}$ denote the rotated inputs of the q/k/v, out, up/gate, and down projections. Values are mean ± std over six seeds.

Adjusted input	PPL (↓)		Avg. Acc (↑)
	Wiki2	C4	0-shot
$\tilde{\mathbf{X}}_{qkv}$	7.957 ± 0.039	12.955 ± 0.060	67.21 ± 0.39
$\tilde{\mathbf{X}}_{ug}$	7.972 ± 0.050	12.964 ± 0.073	67.10 ± 0.59
$\tilde{\mathbf{X}}_{o}$	7.974 ± 0.035	12.988 ± 0.052	66.94 ± 0.69
$\tilde{\mathbf{X}}_{d}$	7.898 ± 0.019	12.869 ± 0.029	**67.74** ± 0.63
$\tilde{\mathbf{X}}_{qkv}$ + $\tilde{\mathbf{X}}_{o}$	7.941 ± 0.034	12.933 ± 0.035	67.46 ± 0.38
$\tilde{\mathbf{X}}_{ug}$ + $\tilde{\mathbf{X}}_{d}$	7.885 ± 0.026	12.830 ± 0.042	67.28 ± 0.43
$\tilde{\mathbf{X}}_{o}$ + $\tilde{\mathbf{X}}_{d}$	7.894 ± 0.032	12.867 ± 0.042	67.61 ± 0.46
All $\tilde{\mathbf{X}}$	**7.876** ± 0.021	**12.809** ± 0.049	67.49 ± 0.21

adjustment only at $\tilde{\mathbf{X}}_{d}$, which gave the highest mean accuracy among the evaluated configurations. Some combinations gave lower perplexity, but none gave higher mean accuracy. The down-projection input also requires an online rotation: replacing the Hadamard transform with a learned dense matrix cannot use the FWHT and would add a full matrix multiplication during inference.

6 Conclusions

We demonstrate that the Hadamard transform, widely used in existing studies to manage activation outliers through linear transformations, can create fragmented distributions when a massive outlier dominates a token's activation. Although such tokens are rare, the fragmentation they cause substantially increases the quantization error of the layer in which they first appear. To address this fragmentation, we propose a massive outlier-aware sign-adjusted rotation method, named SignRot, a computationally efficient method optimized for online transformation. SignRot identifies massive outlier channels and calculates sign adjustment vectors using a single forward pass. Our approach quantizes the LLaMA2-7B model in 3 minutes with RTN and 15 minutes with GPTQ. During inference, SignRot adds only element-wise sign flips per token per decoder layer. On LLaMA3-70B, QuaRot+SignRot achieves a WikiText2 perplexity of 5.88 and an average zero-shot reasoning accuracy of 72.73, surpassing SpinQuant. Our findings suggest that a simple, massive outlier-aware sign adjustment can mitigate this fragmentation of online Hadamard rotation without iterative optimization. Extending SignRot to tokens with several massive outlier channels and to layers whose dominant massive outlier channel varies across tokens (*e.g.,* the last layers in Appendix C), for instance by assigning each channel group its own Hadamard transform and sign adjustment vector, is left for future work.

Acknowledgments and Disclosure of Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25498006). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) under the Leading Generative AI Human Resources Development (IITP-2026-RS-2026-25544527) grant funded by the Korea government(MSIT). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Innovation Human Resources Development (IITP-2026-RS-2026-25547714) grant funded by the Korea government(MSIT). The authors declare no competing interests.

References

[1] Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

[2] Ben Berckmoes and Geert Molenberghs. Approximate central limit theorems. *Journal of Theoretical Probability*, 31(3):1590–1605, 2018.

[3] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[4] 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*, 36:4396–4429, 2023.

[5] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[6] Samuel Rhys Cox and Wei Tsang Ooi. Conversational interactions with npcs in llm-driven gaming: Guidelines from a content analysis of player feedback. In *International Workshop on Chatbot Research and Design*, pages 167–184. Springer, 2023.

[7] Harald Cramér. Über eine Eigenschaft der normalen Verteilungsfunktion. *Mathematische Zeitschrift*, 41(1):405–414, 1936.

[8] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in neural information processing systems*, 35:30318–30332, 2022.

[9] 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.

[10] Fino and Algazi. Unified matrix treatment of the fast walsh-hadamard transform. *IEEE Transactions on Computers*, 100(11):1142–1146, 1976.

[11] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

[12] Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun S Shao, Kurt Keutzer, and Amir Gholami. Kvquant: Towards 10 million context length llm inference with kv cache quantization. *Advances in Neural Information Processing Systems*, 37:1270–1303, 2024.

[13] 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. Mistral 7b, 2023.

[14] Jemin Lee, Sihyeong Park, Jinse Kwon, Jihun Oh, and Yongin Kwon. Exploring the trade-offs: Quantization methods, task difficulty, and model size in large language models from edge to giant. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 2025.

[15] Jun Li, Fuxin Li, and Sinisa Todorovic. Efficient riemannian optimization on the stiefel manifold via the cayley transform. In *International Conference on Learning Representations*, 2020.

[16] Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joseph E. Gonzalez. Train large, then compress: Rethinking model size for efficient training and inference of transformers. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[17] 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. *Advances in Neural Information Processing Systems*, 37:87766–87800, 2024.

[18] 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. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

[19] 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. In *The Thirteenth International Conference on Learning Representations*, 2025.

[20] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2016.

[21] NVIDIA. Nvidia cutlass library, 2023.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[23] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[24] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[25] Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

[26] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[27] Mark F. Schilling, Ann E. Watkins, and William Watkins. Is human height bimodal? *The American Statistician*, 56(3):223–229, 2002.

[28] Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. *arXiv preprint arXiv:2308.13137*, 2023.

[29] Yuantian Shao, Yuanteng Chen, Peisong Wang, Jianlin Yu, Jing Lin, Yiwu Yao, Zhihui Wei, and Jian Cheng. Dartquant: Efficient rotational distribution calibration for LLM quantization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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

[31] Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, et al. Flatquant: Flatness matters for llm quantization. *arXiv preprint arXiv:2410.09426*, 2024.

[32] Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023.

[33] 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.

[34] Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks. *arXiv preprint arXiv:2402.04396*, 2024.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[36] T Wolf. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[37] Jingyang Xiang and Saiqian Zhang. Dfrot: Achieving outlier-free and massive activation-free for rotated llms with refined rotation. *arXiv preprint arXiv:2412.00648*, 2024.

[38] 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 Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 38087–38099. PMLR, 23–29 Jul 2023.

[39] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025.

[40] Zihao Ye. FlashInfer: Kernel Library for LLM Serving. `https://github.com/flashinfer-ai/flashinfer`, 2023.

[41] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, 2019.

A Derivation of the bimodality threshold

This section details the derivation of the bimodality threshold in Equation 3.

Lindeberg condition under RHT. Fix a token t and an output channel j. With independent random signs $r_i \in \{\pm 1\}$, Equation 1 is a sum of independent zero-mean terms $X_i = x_{t,i} r_i h_{i,j}$, where $|X_i| = |x_{t,i}|/\sqrt{C_{in}}$ is deterministic and the total variance is $s_t^2 = \|\mathbf{x}_t\|_2^2 / C_{in}$. Regarding the sum as a row of a triangular array indexed by C_{in}, the Lindeberg condition [2] requires, for every $\epsilon > 0$,

$$\frac{1}{s_t^2} \sum_i \mathbb{E}\big[X_i^2 \, \mathbb{I}\{|X_i| > \epsilon s_t\}\big] = \sum_{i \,:\, x_{t,i}^2 > \epsilon^2 \|\mathbf{x}_t\|_2^2} \frac{x_{t,i}^2}{\|\mathbf{x}_t\|_2^2} \;\to\; 0. \tag{10}$$

Because the indicator is deterministic, the Lindeberg condition holds if and only if $\max_i x_{t,i}^2 / \|\mathbf{x}_t\|_2^2 \to 0$. If the maximum ratio vanishes, the index set becomes empty for any fixed ϵ. Otherwise, choosing ϵ^2 below the limit superior of the maximum ratio keeps the largest term in the sum. If $M_t^2 / \|\mathbf{x}_t\|_2^2$ remains bounded away from zero along the array, the Lindeberg condition fails for the full sum. Since each summand has variance $x_{t,i}^2 / C_{in}$, Feller's negligibility condition fails together with it, so the Lindeberg–Feller converse does not apply and this failure alone does not rule out a Gaussian limit. Non-Gaussianity follows instead from the decomposition: along any subsequence with $M_t^2 / \|\mathbf{x}_t\|_2^2 \to \delta > 0$, the normalized outlier term is an independent symmetric two-point variable with variance δ, and by Cramér's decomposition theorem [7] a sum of independent variables with a non-Gaussian component cannot be Gaussian. A finite magnitude-based massive outlier label alone does not establish this asymptotic property.

Excluding the outlier channel, the residual sum has variance $\sigma_{rest}^2 = \frac{1}{C_{in}} \sum_{i \neq c_{max}} x_{t,i}^2$ and admits a Gaussian approximation when $\max_{i \neq c_{max}} x_{t,i}^2 / \sum_{i \neq c_{max}} x_{t,i}^2 \to 0$. The outlier term

$M_t r_{c_{max}} h_{c_{max},j}$ takes the values $\pm M_t/\sqrt{C_{in}}$ with probability $1/2$ each, independently of the residual sum, which gives the approximate equally weighted mixture in Equation 2.

For a fixed draw of $\mathbf{r}$, a balanced c_{max}-th row of $\mathbf{H}$ assigns the outlier contribution $+M_t/\sqrt{C_{in}}$ to half of the output channels and $-M_t/\sqrt{C_{in}}$ to the other half. Approximating the residuals within each group as Gaussian gives the two-component description of the channel histogram. SignRot aligns the signs of this row, converting these two shifts into a single constant contribution $M_t r_{c_{max}}/\sqrt{C_{in}}$ across all output channels. This is how sign adjustment removes the selected outlier's sign-induced split.

Bimodality of the mixture. Under the two-Gaussian approximation, let $\mu = |M_t|/\sqrt{C_{in}}$, $\sigma = \sigma_{rest} > 0$, and let φ_σ denote the density of $\mathcal{N}(0, \sigma^2)$. The density $f(y) = \frac{1}{2}\varphi_\sigma(y-\mu) + \frac{1}{2}\varphi_\sigma(y+\mu)$ is symmetric about zero, so $f'(0) = 0$. Using $\varphi''_\sigma(y) = (y^2/\sigma^4 - 1/\sigma^2)\,\varphi_\sigma(y)$,

$$f''(0) = \frac{1}{\sigma^2}\left(\frac{\mu^2}{\sigma^2} - 1\right)\varphi_\sigma(\mu). \tag{11}$$

Hence $y = 0$ is a local minimum between two modes if and only if $\mu > \sigma$, and the mixture is unimodal for $\mu \le \sigma$ [27]. Substituting μ and σ gives

$$\frac{|M_t|}{\sqrt{C_{in}}} > \sigma_{rest} \iff M_t^2 > \sum_{i \neq c_{max}} x_{t,i}^2 \iff \rho_t := \frac{M_t^2}{\sum_{i \neq c_{max}} x_{t,i}^2} > 1, \tag{12}$$

where ρ_t denotes the dominance ratio of the outlier channel. Since $\|\mathbf{x}_t\|_2^2 = M_t^2 + \sum_{i \neq c_{max}} x_{t,i}^2$, the condition $\rho_t > 1$ is equivalent to $M_t^2 > \|\mathbf{x}_t\|_2^2/2$, which at most one channel per token can satisfy. Below the threshold, the approximating mixture is unimodal. This threshold is not a necessary or sufficient test for bimodality of an arbitrary empirical activation histogram. In particular, uniqueness of an energy-dominant channel does not exclude multiple sources of fragmentation.

Numerical example. We relate the threshold to the massive outlier token in Table 7, taken from the down-projection input of the 2nd decoder layer of LLaMA2-7B ($C_{in} = 11{,}008$), where $M_t = 1510$. After SignRot, the selected Hadamard row is constant and all other rows are orthogonal to it. Thus, in exact arithmetic, the mean across output channels is $M_t/\sqrt{C_{in}}$ and the population variance across channels is $C_{in}^{-1}\sum_{i \neq c_{max}} x_{t,i}^2 = \sigma_{rest}^2$, using the sign convention of the illustrated token. These identities follow from orthogonality and hold for any input, so the agreement of the reported mean of 14.3906 with $1510/\sqrt{11{,}008} \approx 14.39$ is a consistency check rather than a test of the mixture model. Using the reported variance of 0.3059 gives $\sigma_{rest} \approx 0.553$ and a formal threshold of $\sqrt{C_{in}}\,\sigma_{rest} \approx 58$. The observed outlier is about $26\times$ larger ($\rho_t \approx 677$), accounting for approximately 99.85% of the token's energy. These numerical estimates use the rounded table entries.

This token does not satisfy the residual condition of the two-Gaussian model. The original minimum of -46.16 implies that a single residual channel carries at least $46.16^2/(C_{in} \cdot 0.3059) \approx 63\%$ of the residual energy, and the output after SignRot spans only about $\pm 2.3\,\sigma_{rest}$ around its mean, much narrower than expected for C_{in} Gaussian samples. The separation in Figure 1 does not rely on this approximation. Since the right factor $\mathrm{diag}(\mathbf{n})$ in Equation 7 only flips the signs of output channels of the same transform, each RHT output $|y_{t,j}|$ equals the corresponding output after SignRot, which lies in $[13.1328,\ 15.6328]$ for every j. Hence no channel falls in $(-13.13,\ 13.13)$, and the two clusters are separated by a gap of about 26 without any distributional assumption.

Table 7: Comparison of activation distribution statistics for a token with a massive outlier and a token without an outlier.

	Massive Outlier Token				**Non-outlier Token**			
	Mean	Variance	Min	Max	Mean	Variance	Min	Max
Original	0.1377	207.3750	-46.1562	1510.0000	-0.0000	0.0001	-0.2783	0.5435
After RHT	0.0001	207.3750	-15.6328	15.6250	0.0000	0.0001	-0.0419	0.0378
With SignRot	14.3906	0.3059	13.1328	15.6328	0.0000	0.0001	-0.0419	0.0370

B Effect of sign adjustment on activation distribution

To empirically validate our theoretical analysis of how sign adjustment affects the activation distribution, we analyze the statistics of activations before and after applying SignRot on both a token without massive outliers and a token with a massive outlier from the 2nd decoder layer down projection layer on LLaMA2-7b. As shown in Table 7, for the massive outlier token, the standard Randomized Hadamard Transformation (RHT) reduces the min/max range to [-15.6328, 15.6250], but the variance remains extremely high at 207.3750, indicating that the distribution is still highly dispersed. In contrast, applying SignRot effectively collapses the distribution into a tight cluster with a mean of 14.3906 and a significantly reduced variance of 0.3059, demonstrating that the sign adjustment successfully mitigates the impact of the massive outlier. For the non-outlier token, applying SignRot does not significantly alter the mean, variance, or min/max values compared to RHT, confirming that our method preserves the well-distributed nature of activations when no massive outliers are present.

Table 8: Massive outlier tokens at the down projection input. Tokens/seq is the number of massive outlier tokens per 2048-token sequence; Channels/token is the mean number of massive outlier channels per massive outlier token (for ours, the recursive depth). M_2/M_1 (median) and the energy share of M_1, $M_1^2/\|\mathbf{x}_t\|_2^2$ (median), are over the tokens selected by our criterion. Layers not listed contain no massive outlier token under either criterion.

		Tokens/seq		Channels/token			
Model	Layer	Sun et al. [30]	Ours	Sun et al. [30]	Ours	M_2/M_1	M_1 energy (%)
LLaMA2-7B	2	2.09	0.97	1.52	2.44	0.03	99.8
	31	2.09	0.98	2.00	4.98	0.53	78.1
LLaMA2-13B	4	1.09	1.09	1.00	3.00	0.08	99.0
	39	1.09	1.09	1.00	1.00	0.05	99.3
	40	0.23	0.05	1.20	1.00	0.34	61.2
LLaMA3-8B	2	1.00	1.00	1.00	3.17	0.12	98.5
	32	3.19	0.12	2.07	2.00	0.62	60.7
LLaMA3-70B	4	1.04	1.04	1.00	4.02	0.34	86.9
	79	1.04	0.51	2.67	4.14	0.65	63.1
	80	2.13	0.09	1.32	1.00	0.32	70.0
Mistral-7B-v0.3	2	2.15	2.13	1.76	2.40	0.10	98.9
	27	0.01	0.18	1.00	1.00	0.12	87.8
	28	0.01	0.30	1.00	1.00	0.16	78.5
	30	0.00	2.85	–	1.01	0.24	62.0
	31	2.13	2.10	2.00	2.00	0.68	62.3
	32	0.00	0.41	–	1.00	0.16	54.9

C Statistics of massive outliers and their impact on quantization

We run the FP16 model on 128 WikiText-2 sequences of 2048 tokens and record the input of each down projection layer before rotation. Let $M_1 \geq M_2 \geq \dots$ denote the magnitudes of the largest channels of a token $\mathbf{x}_t \in \mathbb{R}^{C_{in}}$, so that $M_1 = M_t$ in section 3.

Two criteria for a massive outlier token. Sun et al. [30] mark a channel as massive when $|x_{t,c}| > 100$ and $|x_{t,c}| > 1000 \times \text{median}_c|x_{t,c}|$. This criterion identifies large activations, but it is not tied to the condition that causes the bimodal fragmentation in Equation 2. After rotation, a token becomes bimodal only when its largest channel outweighs all other channels combined,

$$\rho_1 = \frac{M_1^2}{\sum_{i \neq c_{max}} x_{t,i}^2} > 1, \tag{13}$$

because the other channels are spread over all C_{in} coordinates and must be compared with M_1 as a whole, not with the per-channel median. We therefore count a token as a massive outlier token when Equation 13 holds and M_1 is at least half of the largest magnitude observed in that layer (layer peak ≥ 100), which excludes low-amplitude sparse tokens. We apply the same test recursively: a token has depth k when $\rho_j = M_j^2 / \sum_{i>j} x_{t,i}^2 > 1$ for $j = 1, \dots, k$, *i.e.,* the j-th channel still dominates

Table 9: Massive outlier tokens at the down projection input of LLaMA2-70B and Qwen3-30B-A3B, with the same columns as Table 8. For Qwen3-30B-A3B, counts are token-expert pairs: each token is routed to 8 of the 128 experts of a layer, and the experts of a layer are pooled.

		Tokens/seq		Channels/token			
Model	Layer	Sun et al. [30]	Ours	Sun et al. [30]	Ours	M_2/M_1	M_1 energy (%)
LLaMA2-70B	3	2.05	1.09	1.00	2.04	0.13	98.4
	9	1.95	0.86	3.05	3.98	0.49	79.1
	13	0.95	0.99	1.00	1.00	0.01	99.9
	60	0.01	0.05	1.00	1.43	0.29	87.5
	62	0.01	0.12	1.00	1.00	0.07	86.6
	64	0.01	0.04	1.00	1.00	0.05	88.7
	65	0.01	0.05	1.00	1.14	0.15	86.3
	66	0.02	0.16	1.00	1.00	0.07	87.0
	72	0.02	1.30	1.00	1.01	0.08	87.3
	73	2.13	1.48	1.00	1.00	0.03	98.6
	74	1.87	2.38	1.00	1.73	0.44	73.4
	76	2.51	2.05	1.00	1.00	0.23	86.3
	77	1.12	1.09	1.97	3.00	0.21	94.2
	78	5.06	3.04	1.60	1.00	0.47	65.3
	79	1.32	0.39	1.35	1.34	0.26	62.6
	80	9.59	0.26	2.58	2.00	0.67	54.2
Qwen3-30B-A3B	2	0.94	0.88	1.00	1.04	0.01	99.9
	3	1.02	1.02	1.00	2.08	0.02	100.0
	4	1.02	1.02	1.00	1.84	0.00	100.0
	42	0.02	0.04	1.00	1.00	0.12	96.0
	43	0.00	2.38	–	1.06	0.26	77.1
	44	0.00	9.72	–	1.05	0.25	73.2
	45	0.00	11.83	–	1.02	0.26	69.7
	46	0.01	36.17	1.00	1.01	0.35	55.6
	47	0.12	8.87	1.00	1.11	0.30	63.5
	48	2.83	726.31	1.06	1.43	0.44	71.3

Table 10: Single-layer quantization KL at the massive outlier layers (GPTQ W4A4). Rank is among all layers of the model with QuaRot; for the 70B models, only 27 layers are measured, and only the first massive outlier layer is listed. Δ is the change from QuaRot to QuaRot+SignRot, over all tokens and over tokens without massive outliers.

Model	Layer	Rank	QuaRot	QuaRot+SignRot	Δ all (%)	Δ others (%)
LLaMA2-7B	2	3	0.0055	0.0021	-61.4	-63.5
	31	1	0.0072	0.0073	$+0.4$	$+4.2$
LLaMA2-13B	4	1	0.0052	0.0033	-36.0	-31.1
	39	3	0.0031	0.0031	$+0.1$	$+0.3$
	40	6	0.0025	0.0025	$+0.5$	$+0.6$
LLaMA3-8B	2	1	0.0255	0.0098	-61.4	-62.4
	32	3	0.0186	0.0186	0.0	$+1.5$
LLaMA2-70B	3	17	0.0017	0.0007	-56.4	-58.9
LLaMA3-70B	4	15	0.0178	0.0069	-61.1	-61.8

after the first $j-1$ are removed, and each additional level splits the rotated distribution once more. We refer to these k channels as the massive outlier channels of the token, ordered as the dominant, second, third, ... massive outlier channel.

The magnitude threshold does not track the split. As shown in Table 8 and Table 9, massive outliers are concentrated in the first few and the last few layers: two to six layers with at most about three tokens per sequence in the models up to 13B and in LLaMA3-70B, and more layers in LLaMA2-70B (16) and Qwen3-30B-A3B (10). In the last layer of Qwen3-30B-A3B, 726 token-expert pairs per sequence pass our criterion. In the first massive outlier layers, the tokens selected by the two criteria share the same largest channel. Elsewhere, the fixed magnitude threshold does not follow the split condition in either direction. It selects tokens that do not split: in the last layer of LLaMA3-8B it selects 3.19 tokens per sequence, of which only 5.1% are bimodal after rotation. It also misses

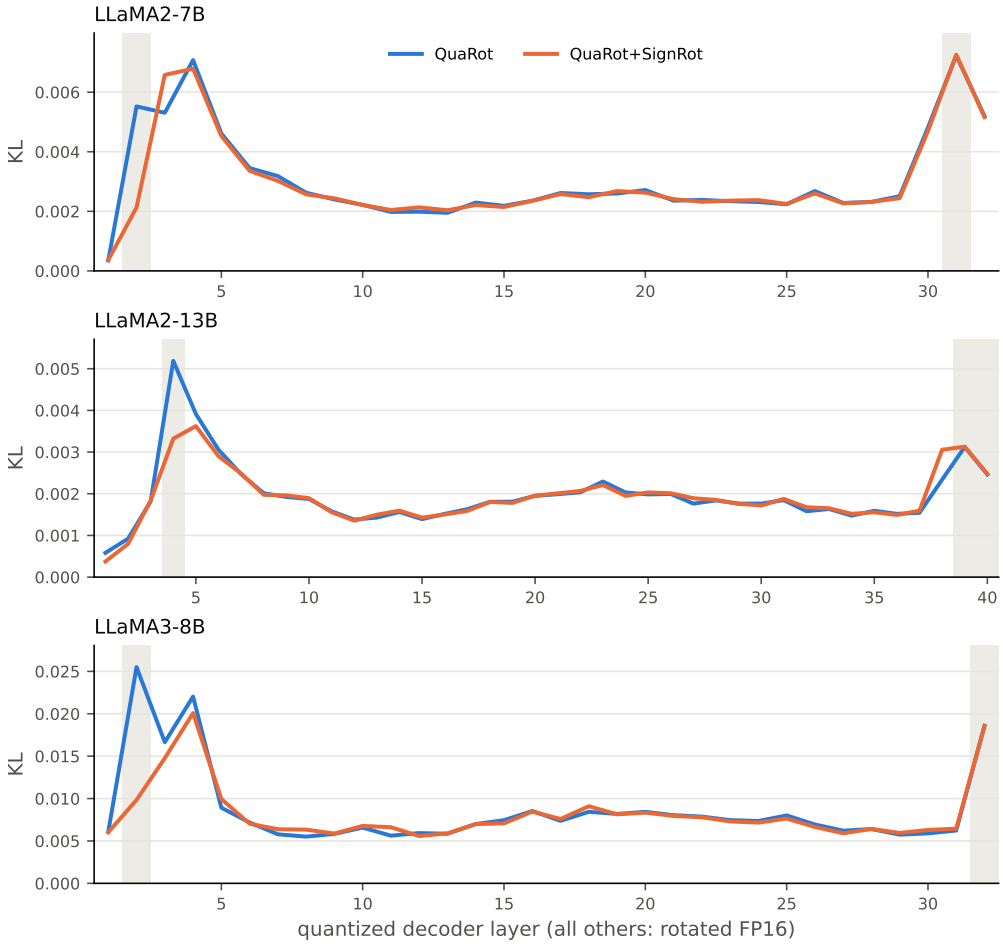


Figure 4: Final KL divergence when only one decoder layer is quantized (GPTQ W4A4, all other layers in rotated FP16). Gray bands mark the layers with massive outlier tokens under our criterion.

tokens that split but stay below the threshold: in layer 30 of Mistral-7B-v0.3 it selects no token, while 2.85 tokens per sequence pass our criterion. The miss is more pronounced when C_{in} is small: in the expert down projection of Qwen3-30B-A3B ($C_{in} = 768$), 2.4–36 token-expert pairs per sequence pass our criterion in layers 43–47, of which the magnitude criterion selects at most 0.12.

The first massive outlier layer. Table 11 lists the massive outlier layers within the first sixth of each network, including LLaMA2-70B and Qwen3-30B-A3B. In every model, the first such layer holds about one massive outlier token per sequence (two in Mistral-7B-v0.3), and all of these tokens share the same largest channel. For all of them, the aligned outlier shift exceeds the largest residual output ($\mu_t > \epsilon_t$, Appendix D), and SignRot narrows their per-token range by 2.6–11.3$\times$ (median).

The largest channel dominates. Under our criterion, a massive outlier token in the first massive outlier layer of the LLaMA models carries 2.0–4.0 massive outlier channels on average, but the second massive outlier channel is small. In the first massive outlier layer of every model except LLaMA3-70B, M_2/M_1 is 0.01–0.13 and M_1 holds 98.4–99.9% of the token energy, so aligning the sign of M_1 alone, as SignRot does, removes almost all of the bimodality. In LLaMA3-70B, M_2/M_1 is 0.34 and M_1 holds 87%; Appendix D shows that SignRot still removes the split of M_1 in this case. The exceptions are the last layers of LLaMA2-7B, LLaMA2-70B, LLaMA3-8B, LLaMA3-70B, and Mistral-7B-v0.3 ($M_2/M_1 = 0.53$–0.68, M_1 holds 54–78%), where a single sign vector cannot fold both channels.

Table 11: Massive outlier layers within the first sixth of each network (down projection input; the first layer of each model in bold). Tokens/seq is per 2048-token sequence; Top channels is the number of distinct largest channels c_{max} over these tokens (per expert for Qwen3-30B-A3B). $\mu_t > \epsilon_t$ and the range ratio (median of the per-token range before over after SignRot) follow Appendix D, measured on 127 WikiText-2 sequences with the sign adjustment vector calibrated on a separate sequence.

Model	Layer	Tokens/seq	Top channels	$\mu_t > \epsilon_t$ (%)	Range ratio
LLaMA2-7B	**2** / 32	0.97	1	100.0	11.34×
LLaMA2-13B	**4** / 40	1.09	1	100.0	7.00×
	3 / 80	1.09	1	100.0	6.05×
LLaMA2-70B	9 / 80	0.87	1	99.1	2.26×
	13 / 80	0.99	1	100.0	9.40×
LLaMA3-8B	**2** / 32	1.00	1	100.0	7.47×
LLaMA3-70B	**4** / 80	1.04	1	100.0	2.63×
Mistral-7B-v0.3	**2** / 32	2.13	1	100.0	5.53×
	2 / 48	0.87	1	100.0	9.51×
Qwen3-30B-A3B	3 / 48	1.02	1	100.0	43.60×
	4 / 48	1.02	1	100.0	885.36×[†]

[†]Nearly one-hot tokens: after SignRot their range is close to zero.

Massive outlier layers dominate the quantization error. To measure how much each layer contributes, we quantize a single decoder layer while keeping all other layers in rotated FP16, and measure the KL divergence of the final output from the FP16 model on WikiText-2 (8 sequences of 2048 tokens, GPTQ W4A4). The KL of quantizing all layers is close to the sum of these single-layer KLs (ratio 0.96, 1.07, and 0.93 for LLaMA2-7B, LLaMA2-13B, and LLaMA3-8B), so the single-layer KL indicates how much each layer contributes. Figure 4 and Table 10 show that the first massive outlier layer is among the three most expensive layers with QuaRot in LLaMA2-7B, LLaMA2-13B, and LLaMA3-8B, at 2.3–3.6× the median of the layers without massive outliers.

SignRot removes the error of the first massive outlier layer. As shown in Table 10, SignRot reduces the single-layer KL of the first massive outlier layer by 36–61%, while the curves of the two methods nearly coincide at layers without massive outliers. The reduction is not limited to the massive outlier tokens themselves: the KL of all other tokens drops by 31–64%. Since SignRot barely changes the KL of layers without massive outliers, this suggests that the error on the massive outlier token, which serves as an attention sink, propagates to every other token through attention in the subsequent layers. At the last layers, SignRot leaves the final KL unchanged, as there is no later layer through which the error of the massive outlier token can reach other tokens. The gain carries over to quantizing all layers. Averaged over random seeds (11 for LLaMA2-7B and LLaMA3-8B, 5 for LLaMA2-13B and Mistral-7B-v0.3), SignRot reduces the KL of the fully quantized model by 2.0–4.5% with GPTQ W4A4 and by 2.9–6.5% with W16A4 in every model, and enabling SignRot only at the first massive outlier layer already yields 1.1–4.8% and 2.3–6.4%, respectively.

D Sign adjustment with multiple massive outlier channels

A massive outlier token can carry more than one massive outlier channel (recursive depth larger than one in Appendix C). We show that SignRot still removes the split induced by the selected channel without any assumption on the other channels. Fix a token t and write Equation 1 as

$$y_{t,j} = s_j\,\mu_t + e_{t,j}, \qquad s_j = \operatorname{sign}(M_t r_{c_{max}} h_{c_{max},j}) \in \{\pm 1\}, \qquad \mu_t = \frac{|M_t|}{\sqrt{C_{in}}}, \tag{14}$$

where the residual $e_{t,j}$ collects every other channel, including the other massive outlier channels, and let $\epsilon_t = \max_j |e_{t,j}|$.

The split of the selected channel is removed for any residual. SignRot multiplies output j by $n_j = \operatorname{sign}(h_{c_{max},j})$ (Equation 6), so $n_j s_j = \operatorname{sign}(M_t r_{c_{max}})$ is the same for all j and the adjusted outputs are $\operatorname{sign}(M_t r_{c_{max}})\,\mu_t + n_j e_{t,j}$. All of them lie in an interval of length $2\epsilon_t$ around a common shift. This uses only $|n_j e_{t,j}| \le \epsilon_t$: the residual may have any shape, and the other massive outlier

channels may still split it, but only within this interval. Before the adjustment, if $\mu_t > \epsilon_t$, no output falls in $(-(\mu_t - \epsilon_t),\ \mu_t - \epsilon_t)$, and the range is at least $2(\mu_t - \epsilon_t)$ whenever the c_{max}-th row contains both signs. Hence the per-token range shrinks by a factor of at least $\mu_t/\epsilon_t - 1$. The energy-dominance condition in Equation 3 therefore keeps its role when a token carries several massive outlier channels: SignRot removes the split of the dominant massive outlier channel, and the remaining channels only enter through ϵ_t. Folding the remaining channels as well, *e.g.,* by assigning each channel group its own Hadamard transform and sign adjustment vector, is left for future work. For example, every massive outlier token in the first massive outlier layer of LLaMA3-70B has more than one massive outlier channel, with $M_2/M_1 \approx 0.34$ (Table 8). Yet SignRot still reduces their per-token range by $2.6\times$ (median over 127 WikiText-2 sequences, with the sign adjustment vector calibrated on a separate sequence).

E Quantization configurations

Table 12: Quantization configurations of the methods in Table 1. Within each pair, the baseline and +SignRot use the same rotation matrices. The clipping ratios of +SignRot are listed in Table 13.

Method	Rotation	Act. quantization	Weight granularity	GPTQ damping
QuaRot	Randomized Hadamard	Per-token symmetric	Per-channel	0.01
QuaRot+SignRot	Randomized Hadamard	Per-token asymmetric	Per-channel	0.01
SpinQuant	Released learned rotation	Per-token asymmetric	Per-channel	0.01
SpinQuant+SignRot	Released learned rotation	Per-token asymmetric	Per-channel	0.01
DartQuant	Trained rotation	Per-token asymmetric	Group 128	0.1
DartQuant+SignRot	Trained rotation	Per-token asymmetric	Group 128	0.1

Table 13: Clipping ratios of +SignRot in Table 1 for the activations, keys, and values.

Method	Act. clip	K clip	V clip
QuaRot+SignRot	0.9[a]	0.9	0.9
SpinQuant+SignRot	0.9	0.9	0.9
DartQuant+SignRot	0.9	1.0	1.0

[a] 1.0 on LLaMA3-70B with GPTQ.

Table 12 lists the quantization configuration of each method in Table 1, and Table 13 lists the clipping ratios of +SignRot. All methods use 4-bit symmetric weight quantization and group-wise asymmetric 4-bit KV cache quantization with a group size of the head dimension. QuaRot follows its official setting [1], and SpinQuant uses the rotation matrices released by its authors [19]. Within each pair, +SignRot uses the same rotation matrices, weight quantization, and GPTQ settings as the baseline; it changes only the activation quantizer to per-token asymmetric quantization and the clipping ratios. QuaRot and QuaRot+SignRot on Mistral-7B-v0.3 and Qwen3-30B-A3B (Table 2) use the same configurations as on the LLaMA models, with an activation clipping ratio of 0.9 for QuaRot+SignRot.

On LLaMA3-70B with GPTQ, QuaRot+SignRot uses an activation clipping ratio of 1.0. This exception comes from QuaRot itself rather than from SignRot: with asymmetric activation quantization and an activation clipping ratio of 0.9, the official QuaRot implementation already degrades severely on this model (WikiText2 perplexity 10.05, compared with 6.60 with symmetric activations). We therefore use 1.0 for activations in the asymmetric QuaRot configurations on LLaMA3-70B with GPTQ, including the asymmetric QuaRot baseline in Table 4.

F Full results of main result table

In Tables 14, 15 and 16, we present the full results of the main result table in the paper, including all evaluated results of LLaMA models. These tables provide a comprehensive comparison of our proposed SignRot method against various baselines such as QuaRot [1], SpinQuant [19], and DartQuant [29], demonstrating the consistent performance improvements achieved by SignRot in managing activation outliers and enhancing quantization efficiency.

Table 14: Performance comparison when applying SignRot to QuaRot. '+SignRot' indicates that our method has been applied.

Model	Method	W Quantizer	Wiki2	C4	ARC-C	ARC-E	HellaSwag	LAMBADA	PIQA	Winogrande	Avg
2-7B	FP16	-	5.47	7.26	46.16	74.54	75.98	73.92	79.05	69.06	69.79
	QuaRot	RTN	8.56	11.86	36.60	61.41	65.07	48.06	72.20	63.06	57.73
	QuaRot+SignRot	RTN	7.93	11.21	38.31	62.84	66.60	58.63	73.61	62.59	60.43
	QuaRot	GPTQ	6.10	8.32	42.32	68.35	72.53	65.40	76.33	65.11	65.01
	QuaRot+SignRot	GPTQ	6.09	8.31	42.83	70.08	73.26	71.73	78.07	67.17	67.19
2-13B	FP16	-	4.88	6.73	49.15	77.44	79.39	76.73	80.47	72.14	72.55
	QuaRot	RTN	6.10	8.67	42.83	69.95	73.54	65.62	77.69	67.88	66.25
	QuaRot+SignRot	RTN	6.00	8.48	45.05	71.68	73.66	69.96	78.18	66.54	67.51
	QuaRot	GPTQ	5.40	7.54	45.48	73.27	76.03	69.01	79.05	70.64	68.91
	QuaRot+SignRot	GPTQ	5.32	7.43	48.29	74.49	77.20	74.40	78.78	71.03	70.70
2-70B	FP16	-	3.32	5.71	57.17	81.02	83.81	79.60	82.70	77.98	77.05
	QuaRot	RTN	4.14	6.42	52.22	76.60	79.96	74.61	81.12	76.32	73.47
	QuaRot+SignRot	RTN	4.05	6.32	52.56	77.36	80.74	76.65	80.58	75.30	73.86
	QuaRot	GPTQ	3.79	6.12	55.46	79.76	81.58	79.35	81.83	76.09	75.68
	QuaRot+SignRot	GPTQ	3.74	6.04	55.20	78.28	82.59	78.88	81.23	77.90	75.68
3-8B	FP16	-	6.14	9.45	53.50	77.57	79.12	75.51	80.74	72.93	73.23
	QuaRot	RTN	10.60	17.19	38.65	66.54	68.82	57.20	71.82	65.04	61.34
	QuaRot+SignRot	RTN	9.68	15.72	42.66	67.80	71.35	64.47	74.27	64.48	64.17
	QuaRot	GPTQ	8.16	13.38	45.73	70.83	72.97	62.70	75.35	67.17	65.79
	QuaRot+SignRot	GPTQ	7.85	12.74	44.88	71.72	74.16	68.83	76.06	67.96	67.27
3-70B	FP16	-	2.86	7.17	64.25	85.94	84.93	79.37	84.44	80.74	79.95
	QuaRot	RTN	55.44	79.48	22.18	34.30	32.15	13.35	57.67	52.49	35.36
	QuaRot+SignRot	RTN	41.47	58.26	22.78	40.74	34.52	20.70	60.77	51.61	38.52
	QuaRot	GPTQ	6.60	12.87	49.49	74.37	77.22	71.69	78.89	71.03	70.45
	QuaRot+SignRot	GPTQ	5.88	11.24	52.65	77.36	79.48	72.09	80.96	73.88	72.73

Table 15: Performance comparison when applying SignRot to SpinQuant. All results are reproduced using SpinQuant author-provided rotation matrices. '+SignRot' indicates that our method has been applied.

Model	Method	W Quantizer	Wiki2	C4	ARC-C	ARC-E	HellaSwag	LAMBADA	PIQA	Winogrande	Avg
2-7B	FP16	-	5.47	7.26	46.16	74.54	75.98	73.92	79.05	69.06	69.79
	SpinQuant	RTN	7.71	11.14	36.77	64.27	66.53	60.31	74.43	61.72	60.67
	SpinQuant+SignRot	RTN	7.69	10.94	37.97	64.48	65.88	60.88	74.76	62.04	61.00
	SpinQuant	GPTQ	6.08	8.25	41.21	68.94	72.95	71.63	77.64	65.27	66.27
	SpinQuant+SignRot	GPTQ	6.09	8.27	42.66	70.24	72.18	72.39	77.31	67.64	67.07
2-13B	FP16	-	4.88	6.73	49.15	77.44	79.39	76.73	80.47	72.14	72.55
	SpinQuant	RTN	6.03	8.42	47.18	74.66	73.32	71.78	77.80	70.88	69.27
	SpinQuant+SignRot	RTN	5.99	8.41	47.53	74.87	73.51	71.05	78.02	69.93	69.15
	SpinQuant	GPTQ	5.39	7.46	46.33	74.58	76.09	74.40	78.45	69.77	69.94
	SpinQuant+SignRot	GPTQ	5.37	7.45	47.53	75.38	76.42	74.69	79.43	71.59	70.84
2-70B	FP16	-	3.32	5.71	57.17	81.02	83.81	79.60	82.70	77.98	77.05
	SpinQuant	RTN	4.02	6.29	55.72	78.79	81.03	77.68	82.05	75.53	75.13
	SpinQuant+SignRot	RTN	4.00	6.29	56.06	79.92	80.36	77.86	81.39	76.16	75.29
	SpinQuant	GPTQ	3.73	6.04	56.66	79.71	81.81	78.73	82.15	74.82	75.65
	SpinQuant+SignRot	GPTQ	3.72	6.04	56.14	79.76	81.45	78.92	81.45	77.03	75.79
3-8B	FP16	-	6.14	9.45	53.50	77.57	79.12	75.51	80.74	72.93	73.23
	SpinQuant	RTN	9.57	15.42	41.38	68.48	71.07	62.74	75.14	66.77	64.26
	SpinQuant+SignRot	RTN	9.41	15.42	41.81	67.76	71.14	63.07	73.99	66.14	63.99
	SpinQuant	GPTQ	7.85	12.71	47.18	71.38	74.71	69.45	76.77	68.27	67.96
	SpinQuant+SignRot	GPTQ	7.87	12.60	47.27	73.23	74.79	72.19	77.75	70.96	69.36
3-70B	FP16	-	2.86	7.17	64.25	85.94	84.93	79.37	84.44	80.74	79.95
	SpinQuant	RTN	24.33	40.84	23.29	39.23	39.12	16.51	57.62	51.54	37.88
	SpinQuant+SignRot	RTN	23.14	37.88	25.17	47.85	45.09	21.73	62.57	53.51	42.65
	SpinQuant	GPTQ	6.38	13.98	53.24	77.86	74.16	70.48	79.11	72.06	71.15
	SpinQuant+SignRot	GPTQ	5.45	10.54	53.75	77.53	79.56	74.64	79.82	75.85	73.52

F.1 Total quantization time

We report the total quantization time comparison in Table 17. RTN+QuaRot+SignRot quantizes the LLaMA2-7B and LLaMA2-70B models in only 3 and 13 minutes, respectively, orders of magnitude faster than SpinQuant [19], which requires hours of multi-GPU training. Even when incorporating GPTQ, SignRot matches the quantization time of QuaRot [1], since the sign adjustment vector $\boldsymbol{n}$ is computed within the GPTQ calibration pass at negligible additional cost.

Table 16: Performance comparison when applying SignRot to DartQuant. All results are reproduced using rotation matrices trained with the DartQuant method. '+SignRot' indicates that our method has been applied.

Model	Method	Wiki2	C4	ARC-C	ARC-E	HellaSwag	LAMBADA	PIQA	Winogrande	Avg
2-7B	DartQuant	5.97	8.07	42.58	69.74	73.49	71.36	77.69	67.01	66.98
	DartQuant+SignRot	5.94	8.04	42.58	70.58	73.83	71.92	77.48	67.25	67.27
2-13B	DartQuant	5.32	7.39	46.59	75.63	77.26	75.02	78.51	69.46	70.41
	DartQuant+SignRot	5.31	7.37	47.35	75.34	76.87	74.64	79.33	70.01	70.59
2-70B	DartQuant	3.66	6.01	54.27	78.20	83.06	79.16	82.21	76.80	75.61
	DartQuant+SignRot	3.66	6.00	54.86	79.00	82.90	78.92	82.10	77.35	75.86
3-8B	DartQuant	7.49	12.16	48.21	73.82	75.40	69.55	76.88	71.74	69.27
	DartQuant+SignRot	7.40	12.05	48.04	71.76	76.07	71.16	76.93	70.24	69.03
3-70B	DartQuant	4.93	9.45	57.94	80.68	82.95	76.17	82.10	78.53	76.39
	DartQuant+SignRot	4.86	9.26	57.76	80.26	83.02	76.17	81.56	77.35	76.02

Table 17: Total quantization time comparison on a single A100 GPU. SpinQuant requires 8×A100 GPUs for its rotation training phase.

Methods	LLaMA2-7B	LLaMA2-13B	LLaMA2-70B
RTN, QuaRot+SignRot	1+2min	2+3min	5+8min
GPTQ	14min	26min	1.8h
GPTQ, QuaRot	1+14min	2+26min	0.1h+1.8h
GPTQ, QuaRot+SignRot	1+14min	2+27min	0.1h+1.8h
GPTQ, SpinQuant	1.3h+0.2h	2.4h+0.4h	12h+1.8h

G Performance analysis of SignRot

Table 18: End-to-end inference latency and peak memory on a single RTX 3090 GPU at batch size 1 (2048-token prefill, 128 decoding steps) for symmetric and asymmetric per-token activation quantization.

Model	Method	Act asym	Prefill (ms)	Decode (ms/token)	E2E (ms)	Peak memory (GB)
2-7B	FP16	-	496.7	24.01	3626.6	13.84
	QuaRot	✗	228.3	14.13	2037.6	4.24
	QuaRot+SignRot	✗	232.9	14.17	2047.3	4.24
	QuaRot	✓	283.4	14.92	2194.0	4.33
	QuaRot+SignRot	✓	287.3	14.97	2203.5	4.33
2-13B	FP16	-		OOM		
	QuaRot	✗	635.9	21.41	3376.5	7.72
	QuaRot+SignRot	✗	643.2	21.56	3405.3	7.72
	QuaRot	✓	722.5	22.46	3600.1	7.72
	QuaRot+SignRot	✓	728.3	22.50	3610.4	7.73

G.1 End-to-end inference latency

Table 18 reports the end-to-end measurements behind Table 3 for both symmetric and asymmetric per-token activation quantization. We run the full model with our deployment kernels: 4-bit matrix multiplications, online Hadamard transforms (with the sign adjustment for SignRot), and a 4-bit paged KV cache decoded with FlashInfer. Weights are quantized with RTN in every run, since latency does not depend on weight values, and each decoding step is replayed from a captured CUDA graph (over a static KV cache for FP16). Each configuration is measured in two interleaved rounds after a 90-second thermal soak, with five timed repeats per round, and we report the mean. With symmetric activation quantization, SignRot adds 2.02% to the prefill latency and 0.48% to the end-to-end latency over QuaRot on LLaMA2-7B, similar to the asymmetric case. On LLaMA2-13B, SignRot adds 0.80% to the prefill latency and 0.29% to the end-to-end latency with asymmetric activation quantization (1.15% and 0.85% with symmetric); the FP16 model does not fit in the 24 GB memory of the RTX 3090.

Table 19: Inference speed of a single linear layer for a 2048 sequence length. The matrix sizes correspond to the down-projection layers of LLaMA2-7B, LLaMA2-13B, and LLaMA2-70B. We averaged over 100 runs and report the numbers in milliseconds. Each '+' indicates the addition of the corresponding method on top of the previous column (*i.e.,* +QuaRot refers to Int4+QuaRot, +SignRot refers to Int4+QuaRot+SignRot)

Layer Size	Batch	FP16	Int4	+QuaRot	+SignRot	Speedup
11008x4096	1	2.59	0.73	0.77	0.85	3.05×
	2	5.06	1.45	1.52	1.65	3.07×
	4	9.85	2.92	3.04	3.29	2.99×
	8	19.78	5.73	5.97	6.44	3.07×
	16	39.71	11.27	11.79	12.76	3.11×
	32	79.58	22.53	23.51	25.39	3.13×
13824x5120	1	4.00	1.02	1.14	1.24	3.23×
	2	7.94	2.08	2.31	2.48	3.20×
	4	15.62	4.17	4.55	4.85	3.22×
	8	31.01	8.26	8.94	9.54	3.25×
	16	61.88	16.37	17.62	18.81	3.29×
	32	123.98	32.61	34.93	37.19	3.33×
28672x8192	1	12.61	2.74	2.77	2.91	4.33×
	2	25.84	5.63	5.63	5.89	4.39×
	4	51.03	11.53	11.42	11.89	4.29×
	8	102.30	23.10	22.96	23.91	4.28×
	16	206.30	46.19	45.53	47.46	4.35×
	32	417.19	92.16	91.50	95.45	4.37×

G.2 Single linear layer latency

To isolate the cost of SignRot, we benchmark the down-projection layer alone, where all of its additional operations are concentrated. Table 19 reports the latency (ms) of Int4+QuaRot with and without SignRot. We benchmark the down-projection layer sizes of LLaMA2-7B (11008×4096), LLaMA2-13B (13824×5120), and LLaMA2-70B (28672×8192) at a sequence length of 2048, averaged over 100 runs. The overhead decreases with larger models, from a maximum of 10.39% on LLaMA2-7B (batch size 1) to a minimum of 4.12% on LLaMA2-70B (batch size 4), since the sign adjustment cost becomes negligible compared to the dominant matrix multiplication. The overhead also diminishes with larger batch sizes as computation becomes the primary bottleneck, and SignRot still achieves a 2.99×–4.39× speedup over FP16 on this layer.

H Limitations

Despite the significant performance improvements of SignRot, there are still some limitations. First, it is unknown how to efficiently adjust the online transformation when there are no massive outliers. Second, further research is needed to determine whether SignRot is effective when using transformation matrices other than the Hadamard matrix. Third, SignRot folds only the channel selected during calibration in each layer. The split induced by a second massive outlier channel remains (Appendix D), and in some later layers massive outliers do not occur at a fixed channel: the largest channel differs from token to token (*e.g.,* in the last layers of Mistral-7B-v0.3 and Qwen3-30B-A3B), so tokens whose largest channel is not the selected one are not folded. Lastly, the efficiency of the Hadamard transformation decreases when the intermediate dimension is small, such as in MoE architectures.

I Broader impact

This paper presents SignRot, a computationally efficient method optimized for online transformation. SignRot can adjust online transformations to manage massive outliers, which cannot be optimized by learning-based methods. By enabling efficient quantization of large language models, SignRot can reduce the computational resources required for training and inference, making advanced AI

technologies more accessible and environmentally sustainable. However, if misused for harmful purposes, it could have a negative social impact.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope. The abstract clearly states the problem being addressed, the proposed method, and the key results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of our work are discussed in Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: This paper provides a theoretical analysis of the fragmentation problem in section 3 and how sign adjustment can mitigate it in subsection 4.2.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: This paper shows the main algorithm in section 4 and provides detailed information on the experimental setting in section 5.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code is publicly available at `https://github.com/HYU-ARK-Lab/SignRot`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The experimental setting is described in section 5.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: As we use large models (with at least 7B parameters), different runs do not have too different outputs. We do not repeat the experiments for the main results, as they are costly for models with up to 70B parameters. For the ablation studies on LLaMA2-7B and LLaMA3-8B, we report results averaged over multiple random seeds (Table 4 and Table 6, the latter with standard deviations) or for each seed (Table 5).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The compute resources used for the experiments are described in section 5.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that our research conforms to it in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the societal impacts of our work in Appendix I.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We develop our code using publicly available libraries, models, and datasets. We release our code under the MIT license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code required to reproduce the main experimental results is publicly available at `https://github.com/HYU-ARK-Lab/SignRot`, and the detailed guide for experiments is included in its README file. The code is released under the MIT license.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The LLM is used only for writing, editing and formatting purposes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.