OffQ: TAMING STRUCTURED OUTLIERS IN LLM QUANTIZATION BY OFFSETTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-bit quantization is now a standard approach for accelerating large language model (LLM) inference, as it substantially reduces memory usage and compute cost. However, structural activation outliers remain a major obstacle to effective quantization. In this paper, we introduce OffQ, a post-training quantization method that mitigates this issue through a novel offsetting mechanism. Specifically, OffQ first identifies a low-dimensional outlier subspace in the activations using a tailored top-1 PCA, and then concentrates high-magnitude activations into a single channel by rotation. It then absorbs this concentrated outlier energy into a shared offset, reducing the variance of the activations and making them more amenable to uniform-grid, uniform-precision quantization. This strategy enables accurate W4A4KV4 quantization of LLMs while preserving efficient low-bit execution. Experiments across diverse LLM architectures and benchmarks show that OffQ consistently outperforms state-of-the-art baselines, improving accuracy while maintaining low-bit efficiency. Code will be released upon acceptance.

1 INTRODUCTION

Quantization (Krishnamoorthi, 2018; Frantar et al., 2023; Ashkboos et al., 2024c; Liu et al., 2025; Saxena et al., 2025) is a key technique for accelerating inference and reducing the memory footprint of large language models (LLMs), whose importance is amplified by the increasing demand to deploy LLMs in resource-constrained environments. By representing weights and activations with low-bit (*e.g.*, 4-bit) integers instead of 16-bit floating-point values, quantization substantially reduces storage, memory traffic, and computational overhead, which directly translates into higher throughput, lower latency, and lower serving cost (Zhao et al., 2024; Lin et al., 2025).

Despite this promise, robust 4-bit quantization of W4A4KV4 (weights, activations, and KV cache all quantized to 4 bits) remains challenging for LLMs. A key obstacle is the presence of activation outliers: sparse but extremely large values in a small number of channels that dominate the dynamic range (Bondarenko et al., 2021; Dettmers et al., 2022). These outliers force a coarse quantization step size that is suboptimal for the majority of activations, leading to large rounding errors. QuaRot (Ashkboos et al., 2024c) and SpinQuant (Liu et al., 2025) apply rotation transformations to redistribute activation energy into Gaussian-like distributions, significantly reducing the impact of outliers. Yet, the resulting activation distribution still retains large variance and remains suboptimal for quantization (Shao et al., 2026), limiting the reliability of W4A4KV4 quantization.

To further improve upon rotation-based methods, one line of work learns more effective rotations or uses more expressive equivalent transformations, *e.g.*, OSTQuant (Hu et al., 2025b), AffineQuant (Ma et al., 2024), DuQuant (Lin et al., 2024a), FlatQuant (Sun et al., 2025), DartQuant (Shao et al., 2026), KurTail (Akhondzadeh et al., 2025), at the cost of additional parameters and computational overhead. A second line of work adopts mixed precision, *e.g.*, SliM-LLM (Huang et al., 2025), Atom (Zhao et al., 2024), ResQ (Saxena et al., 2025), preserving higher precision for outlier-heavy channels while aggressively quantizing the rest. Nevertheless, it introduces heterogeneous execution, extra kernels, additional casting, and memory movement, often undermining the simplicity and efficiency of low-bit inference. A third line of work uses non-uniform quantization, *e.g.*, RCP (Choi et al., 2025), NestQuant (Savkin et al., 2025), which adapts the quantization levels to the activation distribution but requires more complex kernels and lacks broad hardware support.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

By contrast, here, we build on recent progress in singular defect analysis (Wang et al., 2025), which shows that activation outliers exhibit a pronounced *low-dimensional structure*: high-norm tokens in an LLM share the same dominant direction across layers, token positions, and locations within a sequence. We refer to this type of outlier, $x \in \mathbb{R}^{1\times D}$, as *structured outliers*, which can be modeled as

$$x = \mu + \epsilon, \tag{1}$$

where μ is a constant vector representing the outlier component, and ϵ is the non-outlier component with a substantially smaller norm. Another example of structured outliers is the bias term in the self-attention layers of certain LLMs, *e.g.*, Qwen 2.5 (Qwen et al., 2025), where these biases introduce input-agnostic outliers into the KV cache. Appendix A visualizes these structured outlier examples.

We introduce ***OffQ***, a simple, efficient, and effective method for mitigating structured outliers in 4-bit quantization without relying on mixed-precision components, non-uniform quantization levels, or compute- and memory-intensive backpropagation-based rotation learning. The core idea is to eliminate the effect of outliers from the bulk of the activation distribution so that standard low-bit quantization can provably most effectively utilize its representational capacity.

Specifically, OffQ first identifies the low-dimensional outlier subspace via a tailored Top-1 PCA and rotates the activations so that outliers concentrate into as few channels as possible. It then reorders the channels into groups, assigning one outlier channel to each group. Finally, OffQ applies a specialized Hadamard rotation to convert the outlier energy into group-wise offsets, which are absorbed into the zero-point of standard asymmetric low-bit quantization. In essence, the offset within each group removes the corresponding outlier channel, yielding activations with substantially reduced variance that can be quantized much more effectively.

We evaluate OffQ on a range of LLM architectures and benchmarks under the challenging W4A4KV4 setting. Across these experiments, OffQ consistently improves perplexity and accuracy relative to state-of-the-art baselines while maintaining an average effective bit-width of 4.25. We also test end-to-end inference latency. Compared with 16-bit baselines, OffQ achieves 1.5-1.7$\times$ the speed in prefill and 2.0-2.4$\times$ the speed in decode. Overall, OffQ provides a practical mechanism for making 4-bit quantization more robust, strengthening the feasibility of low-cost LLM deployment.

2 Related Work

Quantization-Aware Training and Post-Training Quantization. LLM quantization methods are commonly grouped into quantization-aware training (QAT) (Jacob et al., 2018; Chen et al., 2025; Liu et al., 2024c; Ashkboos et al., 2025; Xu et al., 2024; Wang et al., 2023; Du et al., 2024; Kim et al., 2022; O'Neill & Dutta, 2023) and post-training quantization (PTQ) (Arai & Ichikawa, 2025; Kim et al., 2024a; Wei et al., 2022; Dettmers et al., 2022; Huang et al., 2024; Li et al., 2025; van Baalen et al., 2024; Liu et al., 2024b; Xu et al., 2025). QAT simulates quantization effects during training and allows the model to adapt to low-precision representations, but it can be computationally and memory-wise prohibitive. In contrast, PTQ works directly on pre-trained models and reduces the computational and memory requirements. We focus on PTQ due to its efficiency and practicality.

Weight-Only, Weight-Activation, and KV-Cache Quantization. In PTQ, weight-only quantization (Frantar et al., 2023; Muller et al., 2025; Dettmers et al., 2024; Kim et al., 2024b; Tseng et al., 2024b; Zhang et al., 2024b; Lin et al., 2024b; Hu et al., 2025a; Guo et al., 2024; Zhang & Shrivastava, 2025; Lee et al., 2023; Zhang et al., 2024a; Arai & Ichikawa, 2026; Chee et al., 2024; Tseng et al., 2024a; Huang et al., 2025; Adepu et al., 2024) compresses model weights while keeping activations in higher precision. This reduces memory traffic during decoding, but it still requires full-precision matrix multiplication and does not fully exploit the efficiency of low-bit Tensor Cores. By contrast, weight-activation quantization further reduces compute cost in compute-bound settings such as prefill and batched inference (Ashkboos et al., 2024b; Kurtic et al., 2025), because the arithmetic can be performed in low-bit using Tensor Cores (Lin et al., 2024b; Ashkboos et al., 2024c; Zhao et al., 2024; Thakkar et al., 2023). KV-cache quantization (Su et al., 2025; Cai et al., 2025; Yang et al., 2025; He et al., 2024; Hooper et al., 2024; Liu et al., 2024d; Son et al., 2025; Zandieh et al., 2026) reduces the memory bandwidth overhead of autoregressive decoding by storing keys and values in low precision. In this work, we focus on the challenging but practically valuable W4A4KV4 setting, where weights, activations, and the KV cache are all quantized to 4 bits.

Equivalent Transforms in Weight-Activation Quantization. A central challenge in weight-activation quantization is the presence of outliers in activations. Existing solutions mainly rely on equivalent transforms: applying a linear transformation to activations before quantization and its inverse after dequantization, which redistribute the activation energy to be less spiky and more quantization-friendly. Prior work has explored several kinds of equivalent transforms:

(i) *Permutation.* RPTQ (Yuan et al., 2023) permutes channels so that similar value ranges are clustered into groups and then applies fine-grained quantization.

(ii) *Scaling and shifting* are used in SmoothQuant (Xiao et al., 2023), Outlier Suppressions (Wei et al., 2022; 2023), OmniQuant (Shao et al., 2024), and ABQ-LLM (Zeng et al., 2025).

(iii) *Rotation.* Rotation was initially used as an equivalent transform in SliceGPT (Ashkboos et al., 2024a) for model trimming, and later became a widely used technique for outlier suppression in quantization. QuaRot (Ashkboos et al., 2024c) uses random Hadamard rotations; SpinQuant (Liu et al., 2025) trains rotations with Cayley SGD; DartQuant (Shao et al., 2026) optimizes rotations using the Whip loss; DFRot (Xiang & Zhang, 2025) learns rotations with a weighted loss; KurTail (Akhondzadeh et al., 2025) learns rotations with a kurtosis-based objective; DuQuant (Lin et al., 2024a) combines two rotations with permutation; ResQ (Saxena et al., 2025) uses rotation and retains $1/8$ of the channels at higher precision; Both NestQuant (Savkin et al., 2025) and RCP (Choi et al., 2025) combine rotation with non-uniform grids.

(iv) *Affine transform.* AffineQuant (Ma et al., 2024) and FlatQuant (Sun et al., 2025) learn affine transforms with an MSE objective; QLLM (Liu et al., 2024a) reallocates outliers across channels; OSTQuant (Hu et al., 2025b) and QServe (Lin et al., 2025) combine rotation and scaling.

In this work, we introduce a new equivalent transform:

(v) *Outlier2offset*, which converts outliers into offsets. This technique is complementary to existing methods, including backpropagation-based rotation learning, mixed precision, and non-uniform quantization grids, and can be combined with them for further improvement. Nevertheless, we show that OffQ achieves strong performance under the standard uniform-grid, uniform-precision setting, leaving the exploration of advanced combinations for future work.

3 Background and Motivation

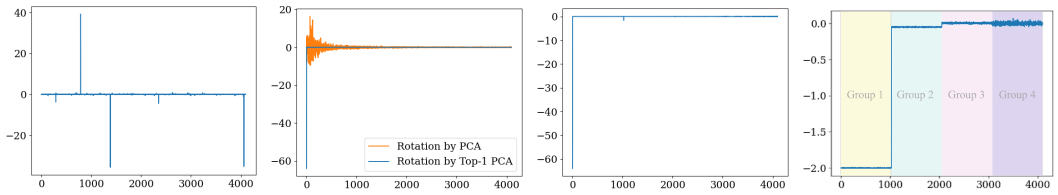
Quantization maps high-precision floating-point values to lower-bit discrete representations, reducing memory footprint and accelerating inference. We adopt a *uniform quantizer* (Krishnamoorthi, 2018), which transforms a floating-point value x into a b-bit integer x_q via

$$x_q = \text{clamp}\left(\left\lfloor \frac{x}{s} \right\rceil + z, 0, 2^b - 1\right), \tag{2}$$

where $\lfloor \cdot \rceil$ denotes the rounding operator, s is the scale factor defining the quantization step size, and z is the zero-point, an integer offset that enables *asymmetric quantization*. Omitting z yields *symmetric quantization*. The granularity of quantization parameters can vary. *Per-tensor quantization* uses a single pair (s, z) for an entire weight or activation tensor; although memory-efficient, it is highly sensitive to outliers. *Per-channel quantization* assigns separate parameters to each output channel and captures channel-wise distributional variance more effectively. *Per-group quantization* further partitions each channel into groups with dedicated parameters, offering finer-grained fidelity at the cost of additional metadata storage.

To address outlier channels, rotation-based methods rotate the outlier token $x \in \mathbb{R}^{1\times D}$ with an orthonormal matrix $R \in O(D)$. The resulting vector becomes more Gaussian-like, and the extreme values are suppressed. However, when activations contain structured outliers of the form $x = \mu + \epsilon$, where μ is a high-norm constant vector and $\|\epsilon\| \ll \|\mu\|$, a better rotation can be constructed.

Without loss of generality, assume that $\epsilon \sim \mathcal{N}(0, I_D)$. We seek a rotation $R \in O(D)$ that minimizes the range spanned by the coordinates of $y = xR$, namely $\max_i y_i - \min_i y_i$, since this range determines the granularity of quantization levels. Denote the i-th column of R as R_i, and let $t_i = R_i \cdot \mu$, where $\cdot$ is inner product. Then $y_i = x \cdot R_i = (R_i \cdot \mu + R_i \cdot \epsilon) \sim \mathcal{N}(t_i, 1)$. Now imagine we have a pistol and shoot on a line. Assume the bullet dispersion follows $\mathcal{N}(t_i, 1)$ when targeting t_i. We shoot D times independently, each aiming at target t_i. It is obvious that to minimize the range spanned by these D bullet holes, we must aim at the same target for all shots. So, the optimal solution is $t_1 = t_2 = \cdots = t_D$. In OffQ, we construct a rotation matrix that satisfies this optimality condition.



(a) Token $\boldsymbol{X} \in \mathbb{R}^{1\times 4096}$ (b) Concentrated $\boldsymbol{X}_{\text{rot}}$ (c) Permute to 4 groups (d) Group-wise offset $\boldsymbol{X}_{\text{had}}$

Figure 1: Visualization of the outlier concentration of the top-1 PCA and the subsequent offsetting technique. We plot the activation values of an outlier token in Llama 3-8B before and after each step, where the x-axis is the channel index and the y-axis is the activation value. The top-1 PCA effectively concentrates the outlier into the first channel, while the regular PCA fails to do so due to interference from normal tokens. The permutation and group-wise offsetting then redistributes the outlier value across all channels, creating constant offsets that can be absorbed during quantization.

4 METHOD

We construct the optimal rotation in two steps: In Section 4.1, we find the direction of the structural outlier by Top-1 PCA and concentrate it to the first channel. Then in Section 4.2, we construct the optimal rotation matrix by converting outliers to offsets. Figure 1 illustrates the whole process of outlier-to-offsets. The end-to-end integration of OffQ into LLMs is detailed in Section 4.3.

4.1 CONCENTRATE STRUCTURED OUTLIERS VIA TOP-1 PCA

Top-1 PCA. To extract the low-dimensional outlier subspace, we propose a *top-1 PCA* procedure. Given a small calibration dataset of N sequences each of length L, we collect input activations before each linear layer with hidden dimension D, forming an $N \times L \times D$ tensor. To focus on outlier tokens and suppress interference from normal tokens, we select the top-1 token per sequence with the largest L^∞ norm (see Appendix B for the rationale), yielding a matrix $\boldsymbol{X} \in \mathbb{R}^{N\times D}$. Assuming zero-mean activations, we compute the covariance matrix $\boldsymbol{C}$ and perform eigendecomposition:

$$\boldsymbol{C} = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)/N = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top, \tag{3}$$

where the columns of $\boldsymbol{U}$ are the eigenvectors sorted by their eigenvalues in $\boldsymbol{\Lambda}$ in descending order. The leading eigenvector (first column of $\boldsymbol{U}$) captures the direction of structured outlier μ.

Concentrate Outliers. To align the outlier directions with individual channels, we rotate the $\boldsymbol{X}$ as

$$\boldsymbol{X}_{\text{rot}} = \boldsymbol{X}\boldsymbol{U}. \tag{4}$$

The first channel of $\boldsymbol{X}_{\text{rot}}$ corresponds to the projection onto the leading eigenvector, thereby concentrating the structured outlier μ into the first channel. More generally, the leading G channels capture the dominant outlier directions of the activations, while the remaining channels are largely outlier-free and thus amenable to low-bit quantization.

4.2 ABSORBING OUTLIERS BY OFFSETTING

Offsetting Technique. We construct a partially random Hadamard matrix $\boldsymbol{H} \in \mathbb{R}^{D\times D}$, an orthogonal matrix with ± 1 entries, and whose first row is constrained to be all ones, *i.e.*, $\boldsymbol{H}_{1,:} = [1, 1, \ldots, 1]$. We then apply $\boldsymbol{H}$ to the outlier-concentrated activations $\boldsymbol{X}_{\text{rot}}$,

$$\boldsymbol{X}_{\text{had}} = \boldsymbol{X}_{\text{rot}}\boldsymbol{H}/\sqrt{D}. \tag{5}$$

This Hadamard rotation redistributes the value of the first channel *uniformly* across all channels, creating a constant offset of $1/\sqrt{D}$ times the original outlier value into every channel. To illustrate this, consider a 2-dimensional token $\boldsymbol{X}_{\text{rot}} = [x_1, x_2] \in \mathbb{R}^{1\times 2}$ with an outlier in the first channel x_1. Applying the Hadamard rotation with $\boldsymbol{H} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ gives:

$$\boldsymbol{X}_{\text{had}} = \boldsymbol{X}_{\text{rot}}\boldsymbol{H}/\sqrt{2} = \left[\frac{x_1 + x_2}{\sqrt{2}}, \frac{x_1 - x_2}{\sqrt{2}}\right] = \underbrace{\frac{x_1}{\sqrt{2}}}_{\text{offset}} + \underbrace{\left[\frac{x_2}{\sqrt{2}}, -\frac{x_2}{\sqrt{2}}\right]}_{\text{without outlier } x_1}. \tag{6}$$

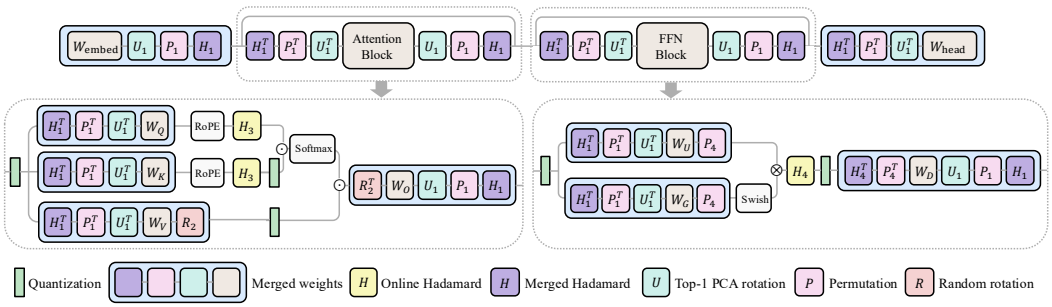


Figure 2: The overall quantization pipeline of OffQ. U_1 is the shared rotation from top-1 PCA on attention and feedforward inputs, P_1 and P_4 are the permutation matrices for grouping, R_2 is a per-head random rotation, H_1 is the group-wise Hadamard rotation for offsetting, and H_3, H_4 with yellow background are the online Hadamard rotations. Most rotation matrices, except for the three online Hadamard rotations, are fused into the weight matrices to reduce computational overhead.

The outlier x_1 is spread across all channels as a constant offset $x_1/\sqrt{2}$, which can be absorbed into the zero-point under asymmetric quantization. The remaining activations exhibit a flatter distribution and can be quantized effectively. The Hadamard rotation here satisfies the optimality condition in Section 3: structured outlier μ is rotated to be in parallel with an all-1 vector. Note that the optimality holds with any orthogonal matrix whose first row contains only one unique value. This enables the combination of the offsetting technique with rotation learning (Liu et al., 2025; Shao et al., 2026).

Group-Wise Offsetting. To absorb multiple outlier channels, we extend the offsetting technique to a group-wise setting. We partition the D channels of $\boldsymbol{X}_{\text{rot}}$ into G groups of size D/G, and reorder the channels so that the first channel of the g-th group corresponds to the g-th largest outlier channel. The offsetting technique is then applied independently within each group, so that the top-G outlier channels are each absorbed as a zero-point under per-group asymmetric quantization. The remaining non-outlier channels are assigned to groups in sorted order similar to (Yuan et al., 2023), which we found empirically to outperform the zigzag assignment of (Lin et al., 2024a).

4.3 Quantizing an LLM with Activation Offsetting

Building upon our insights into the structured nature of outliers and the offsetting technique, we introduce *OffQ*, a novel quantization method that effectively mitigates outlier-induced quantization error. The overall pipeline, illustrated in Figure 2, enhances the architecture of the rotation-based method SpinQuant (Liu et al., 2025) by incorporating our activation offsetting technique.

Step 1. Collect Activation Statistics. We begin by fusing the RMSNorm scaling parameters into the adjacent weight matrices. Activation statistics are then collected at the following positions: (1) *Attention and feedforward inputs.* Input activations to $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V, \boldsymbol{W}_U, \boldsymbol{W}_G$ are stacked together over all layers. The rotation $\boldsymbol{U}_1$ is computed via Top-1 PCA on these activations and shared among layers. (2) *Key activations.* If the LLM contains bias-induced outliers in self-attention, we additionally collect the output activations of $\boldsymbol{W}_K$ after RoPE encoding. The corresponding rotation $\boldsymbol{U}_3$ is computed via per-layer, per-head Top-1 PCA. (3) *Down-projection inputs.* We collect the per-channel absolute maxima of input activations to $\boldsymbol{W}_{\text{D}}$ separately per layer.

Step 2. Apply Rotation, Grouping, and Offsetting. These operations can be represented by orthonormal matrices and are fused into the weight matrices when possible to reduce the overhead.

(1) *Attention and feedforward inputs.* We right-multiply $\boldsymbol{W}_O$ and $\boldsymbol{W}_D$ by $\boldsymbol{U}_1$, apply the permutation $\boldsymbol{P}_1$ to reorder channels into groups, and apply the group-wise Hadamard rotation $\boldsymbol{H}_1$ for offsetting. The inverse transformation $\boldsymbol{H}_1^\top \boldsymbol{P}_1^\top \boldsymbol{U}_1^\top$ is left-multiplied onto $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V, \boldsymbol{W}_U, \boldsymbol{W}_G$ to preserve the original output space. The input embeddings are also rotated by $\boldsymbol{U}_1\boldsymbol{P}_1\boldsymbol{H}_1$, with the inverse applied to the left of the output head $\boldsymbol{W}_{\text{head}}$.

(2) *Value cache.* A per-head random rotation $\boldsymbol{R}_2$ is fused into the right side of $\boldsymbol{W}_V$ and the left side of $\boldsymbol{W}_O$, enabling quantization of the value cache.

Table 1: Compare the perplexity (PPL ↓) on WikiText and the average 0-shot accuracy (0-shot[8] ↑) on ARC-e/ARC-c, BoolQ, HellaSwag, OpenBookQA, PIQA, SIQA, and WinoGrande under W4A4KV4 quantization. Results for 16-bit, GPTQ, QUIK, QuaRot, SpinQuant, and ResQ are referenced from Saxena et al. (2025); and results for DFRot, KurTail, and OSTQuant are referenced from their respective papers. Missing results are marked with "–". Detailed results are provided in Appendix D.

	Method	3-8B		3-70B		3.2-1B		3.2-3B		2-7B		2-13B	
		PPL	0-shot[8]	PPL	0-shot[8]	PPL	0-shot[8]	PPL	0-shot[8]	PPL	0-shot[8]	PPL	0-shot[8]
Llama	16-bit	6.1	67.09	2.9	73.09	9.8	54.86	7.8	62.73	5.5	64.15	4.9	66.45
	GPTQ	166.3	39.79	11655	34.90	108.9	37.98	178.3	40.34	9600	38.89	3120	35.83
	QUIK	14.2	51.60	8.0	58.15	21.8	44.30	15.8	48.74	7.5	56.99	6.8	60.21
	QuaRot	7.8	62.10	5.7	67.56	14.3	49.01	10.1	56.06	6.1	60.75	5.4	63.80
	SpinQuant	7.4	63.76	6.2	65.68	13.6	48.78	9.2	57.89	6.0	60.98	5.2	64.81
	DFRot	7.91	62.35	5.03	68.98	–	–	–	–	6.25	60.97	5.43	63.83
	KurTail	7.2	64.63	4.2	70.69	12.9	50.11	9.0	59.04	5.9	61.31	5.2	65.18
	OSTQuant	7.29	64.70	4.01	**71.16**	–	–	–	–	5.91	**62.11**	5.25	64.19
	ResQ	7.1	63.91	4.1	71.14	12.4	50.11	8.8	58.99	5.8	61.95	**5.1**	**65.25**
	OffQ	**6.98**	**65.49**	**3.88**	70.63	**12.32**	**50.91**	**8.78**	**60.80**	**5.77**	61.99	5.11	**65.25**

	Method	1.5B		3B		7B		14B		32B		72B	
		PPL	0-shot[8]	PPL	0-shot[8]	PPL	0-shot[8]	PPL	0-shot[8]	PPL	0-shot[8]	PPL	0-shot[8]
Qwen 2.5	16-bit	9.3	60.85	8.0	63.81	6.8	68.45	5.3	70.61	5.0	70.44	3.9	73.41
	GPTQ	25770	35.21	9978	35.10	13594	34.85	5100	36.93	3891	38.53	37967	34.54
	QUIK	6614	35.81	15.5	51.19	260.3	41.48	10.5	57.66	9.6	59.08	8.3	61.90
	QuaRot	6600	38.33	68.8	47.76	4036	38.36	6.8	67.14	6.1	67.90	4.9	70.28
	ResQ	12.5	55.26	9.0	61.13	8.2	65.29	6.2	69.16	5.6	69.55	4.6	71.98
	OffQ	**11.35**	**57.53**	**8.98**	**61.47**	**7.66**	**66.16**	**6.07**	**69.20**	**5.52**	**69.59**	**4.29**	**72.68**

(3) *Key cache.* If no bias-induced outliers are present, we apply the per-head online Hadamard rotation $\boldsymbol{H}_3$ to the RoPE-encoded outputs of $\boldsymbol{W}_Q$ and $\boldsymbol{W}_K$. Otherwise, $\boldsymbol{H}_3$ is replaced by $\boldsymbol{U}_3\boldsymbol{P}_3\boldsymbol{H}_3$, which prepends a per-layer, per-head top-1 PCA rotation $\boldsymbol{U}_3$ and a permutation $\boldsymbol{P}_3$.

(4) *Down-projection inputs.* Because rotation cannot penetrate the element-wise multiplication preceding $\boldsymbol{W}_D$, top-1 PCA is not applicable here. Instead, outlier channels are identified via per-channel absolute maxima. Channels are permuted into groups by $\boldsymbol{P}_4$, and then offset by online group-wise Hadamard rotation $\boldsymbol{H}_4$. The permutation $\boldsymbol{P}_4$ is fused into the right side of $\boldsymbol{W}_U, \boldsymbol{W}_G$, and the inverses $\boldsymbol{H}_4^\top \boldsymbol{P}_4^\top$ are fused into the left side of $\boldsymbol{W}_D$.

Step 3. Quantize Weights. Following the practice of rotation-based quantization methods such as QuaRot (Ashkboos et al., 2024c), SpinQuant (Liu et al., 2025), DartQuant (Shao et al., 2026), and KurTail (Akhondzadeh et al., 2025), we apply GPTQ (Frantar et al., 2023) to quantize the weights in 4 bits after merging of transforms into the weight matrices. More advanced weight quantizers such as GPTAQ (Li et al., 2025), QEP (Arai & Ichikawa, 2025), LeanQuant (Zhang & Shrivastava, 2025), Qronos (Zhang et al., 2026), and ResComp (Li et al., 2026) are compatible with our pipeline, but we use GPTQ for all methods to ensure a fair comparison.

5 Experiments

Models and Metrics. We evaluate OffQ on several widely used LLM families, including Llama 2 (7B/13B) (Touvron et al., 2023), Llama 3 (8B/70B) (Grattafiori et al., 2024), Llama 3.2 (1B/3B) (Meta AI, 2024), and Qwen 2.5 (1.5B/3B/7B/14B/32B/72B) (Qwen et al., 2025). Model quality is assessed from two complementary perspectives: language-modeling quality, measured by perplexity (PPL) on the WikiText test split (Merity et al., 2017), and common-sense reasoning, measured by zero-shot accuracy on ARC-e/ARC-c (Clark et al., 2018), BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), and WinoGrande (Sakaguchi et al., 2019).

Table 2: Ablation studies on Llama 3-8B. The table shows the impact of each design choice on WikiText perplexity (PPL ↓) and the zero-shot accuracy (↑) across 8 common-sense reasoning tasks.

Ablation	PPL	0-shot								
	Wiki	ARC-c	ARC-e	BoolQ	HellaS	OBQA	PIQA	SIQA	WinoG	**Avg.**
OffQ	6.98	50.68	77.44	80.43	76.96	43.80	78.89	45.96	69.77	65.49
Without Top-1 Selection	8.27	47.78	73.99	77.95	74.24	41.20	77.58	43.09	69.22	63.13
Zigzag Grouping	7.03	49.40	75.38	79.72	76.30	43.20	79.00	45.29	71.27	64.95
Partial Random Rotation	7.00	47.53	74.33	79.33	76.40	42.00	79.60	45.80	70.24	64.40

Baselines. We compare OffQ against recent state-of-the-art rotation-based post-training quantization methods and their variants, including QuaRot (Ashkboos et al., 2024c), SpinQuant (Liu et al., 2025), DFRot (Xiang & Zhang, 2025), KurTail (Akhondzadeh et al., 2025), OSTQuant (Hu et al., 2025b), and ResQ (Saxena et al., 2025). Specifically, SpinQuant, DFRot, and KurTail optimize rotations with backpropagation; OSTQuant augments rotation with channel-wise scaling, yielding an affine transformation; and ResQ retains high precision for $1/8$ of the channels, resulting in an effective bit-width of 4.5. For reference, we also include two representative PTQ baselines: the weight-only method GPTQ (Frantar et al., 2023) and the mixed-precision method QUIK (Ashkboos et al., 2024b). In addition, we compare with DuQuant (Lin et al., 2024a) and FlatQuant (Sun et al., 2025), whose evaluation protocals are different from ours, in Appendix C.

Implementation and Settings. We implement OffQ using HuggingFace Transformers (Wolf et al., 2020) and PyTorch (Paszke et al., 2017), building on the open-source codebase of ResQ. For activations, OffQ uses per-group asymmetric quantization with a group size of 128, whereas the baselines follow their official implementations and use per-token asymmetric quantization. Baselines with per-group activation quantization are reported in Table 3. Weights are quantized with per-channel symmetric quantization across all methods. Following the evaluation protocol of Saxena et al. (2025), all methods except OSTQuant additionally apply GPTQ for weight quantization. We use 512 samples from the WikiText training split with sequence length 2048 to calibrate Top-1 PCA and 128 samples to calibrate GPTQ. We focus on W4A4KV4, a practically relevant and deployment-friendly setting that reduces memory footprint, bandwidth consumption, and compute cost.

5.1 Quantization Results

Table 1 summarizes results across ten LLMs of various sizes from both the Llama and Qwen families. We make the following observations. (1) Weight-only methods, exemplified by GPTQ, degrades severely under the W4A4KV4 setting: PPL increases from 6.1 to 166.3 on Llama 3-8B and from 6.8 to 13,594 on Qwen 2.5-7B. (2) The mixed-precision method QUIK substantially improves over GPTQ, yet still suffers from significant degradation. (3) Rotation-based methods achieve considerably better performance overall. However, QuaRot fails on Qwen models smaller than 14B, suggesting that naive rotation alone is insufficient to suppress quantization error in the presence of severe outliers in small models. (4) Among rotation-based methods evaluated on the Llama family, learned-rotation approaches such as SpinQuant and KurTail improve over QuaRot but still exhibit a noticeable gap relative to the 16-bit baseline. OSTQuant, which augments rotation with an additional scaling transformation, and ResQ, which enhances rotation with mixed-precision, close this gap further. (5) Our proposed OffQ achieves the best performance across nearly all models and benchmarks, with PPL closer to the full-precision baseline and zero-shot accuracy on par with or exceeding that of the best prior methods, without relying on mixed-precision computation or learned rotation.

5.2 Ablation Study

We conduct ablation studies to analyze the contribution of each component in OffQ. Starting from the full OffQ configuration, we modify each design choice individually. The results are summarized in Table 2, and we discuss the key findings from each ablation below.

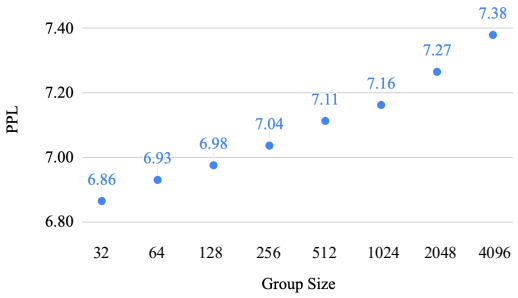


Figure 3: Effect of group size on OffQ using Llama 3-8B with a hidden dimension of 4096. Smaller group size has lower perplexity as more outlier channels can be absorbed by offsetting.

Table 3: PPL of baselines under per-group activation quantization with group size 128. A finer group improves the PPL of baselines, but they are still outperformed by OffQ. Baselines without code release or without per-group activation quantization support are not shown.

Method	PPL	
	Llama2-7B	Llama3-8B
QuaRot	6.02	7.40
SpinQuant	5.87	7.22
DFRot	5.95	7.26
OSTQuant	5.87	7.15
OffQ	5.77	6.98

Top-1 PCA *vs*. PCA. While prior work has applied PCA to identify the outlier subspace for quantization (Saxena et al., 2025), our top-1 PCA is more effective at concentrating outliers by focusing exclusively on the most extreme tokens, which are the primary drivers of quantization error. Because activations are L^2-normalized, the covariance matrix computed over all tokens is distracted by the large number of normal tokens, causing the leading eigenvector to drift from the true outlier direction. Figure 1b validates this intuition, showing that top-1 PCA yields better outlier concentration than standard PCA. Comparing row 1 (with top-1 selection) and row 2 (without top-1 selection) of Table 2, top-1 PCA substantially outperforms standard PCA with all tokens (PPL: 6.98 *vs*. 8.27; 0-shot[8]: 65.49 *vs*. 63.13). This confirms that accurately identifying the outlier subspace, which requires focusing on the most extreme outlier tokens, is critical for the offsetting technique.

Sorted *vs*. Zigzag Grouping. We compare the two strategies for grouping non-outlier channels. Suppose we have 6 values $[1, 2, 3, 4, 50, 60]$ with 2 outliers and 2 groups, sorted grouping yields $[60, 1, 2 \mid 50, 3, 4]$, while zigzag grouping yields $[60, 1, 3 \mid 50, 2, 4]$. We vary the construction of permutation matrices P_1 and P_4 according to the two grouping strategies and find that zigzag grouping (row 3 of Table 2) performs slightly worse than the sorted grouping in OffQ (PPL: 7.03 *vs*. 6.98; 0-shot[8]: 64.95 *vs*. 65.49), likely because sorted grouping clusters channels of similar variance, thereby reducing quantization error.

Hadamard *vs*. Partially Random Rotation. We constructed the partially random Hadamard matrix with all-one in the first row to achieve the offsetting effect, but it can alternatively be replaced by a general partially random rotation matrix whose first row is constant at $1/\sqrt{D}$. On Llama 3-8B, substituting H_1 with such a partially random rotation matrix (row 4 of Table 2) yields comparable perplexity but lower zero-shot accuracy compared with OffQ (PPL: 7.00 *vs*. 6.98; 0-shot[8]: 64.40 *vs*. 65.49), suggesting the Hadamard structure provides a marginal benefit.

Number of Groups. Figure 3 shows the effect of group size on quantization performance. We vary the group size for attention, feedforward, and down-projection inputs, while fixing the group size for the KV-cache to match the head dimension. As expected, more groups allow more outlier channels to be absorbed, reducing quantization error and perplexity; however, finer grouping also increases the number of quantization parameters (scales and zero-points) that must be stored. We select a group size of 128 for activations (average bit-width of 4.25) as it strikes a good balance between quantization performance and storage overhead. Considering that a finer-grained group might also benefit baselines, we apply the per-group (group size 128) asymmetric activation quantization to QuaRot, SpinQuant, DFRot, and OSTQuant in Table 3. We find that OffQ is superior to baselines even when they are additionally equipped with the same per-group activation quantization.

Impact of Calibration. In Table 1, we reported the PPL over the WikiText test split while calibrating using its train split. Here we also calibrate OffQ with C4 (Raffel et al., 2020), PTB (Marcus et al., 1993), and Alpaca (Taori et al., 2023). The resulting Llama3-8B PPL over the WikiText test split is $6.98, 6.95, 6.98$, respectively. Therefore, OffQ is robust to the calibration data.

Table 4: Inference speed of OffQ using vLLM. Time-to-first-token (TTFT) and Time-per-output-token (TPOT) are measured using Llama3-8B on an A100 GPU. We fix the prompt length to 512 and vary the batch sizes from 1 to 64. Here, 16bit is the W16A16KV16 baseline, while OffQ uses W4A4KV4. The suffix g256 and g128 indicate the activation quantization group sizes, and g∞ means that only one group per token, effectively becoming the per-token quantization setting as in QuaRot. Compared to 16bit, OffQ-g128 is 1.5-1.7$\times$ the speed in prefill, and 2.0-2.4$\times$ the speed in decode.

BS	Prefill (TTFT, ms)				Decode (TPOT, ms)			
	16bit	OffQ-g∞	OffQ-g256	OffQ-g128	16bit	OffQ-g∞	OffQ-g256	OffQ-g128
1	38.9	20.3	23.0	25.8	12.6	5.5	5.3	5.3
4	125.7	53.4	63.2	74.9	13.1	5.7	5.5	5.5
16	488.7	206.7	255.2	302.9	14.0	6.9	6.7	6.7
32	979.8	410.1	518.0	609.8	16.4	7.3	7.0	7.1
64	1952.8	818.0	1046.9	1204.6	19.7	9.0	9.0	9.2

5.3 Efficiency of Per-Group Asymmetric Activation Quantization.

We implement the GEMM and quantize kernels in OffQ based on the humming kernels (vLLM Project, 2026), and then integrate them into vLLM (Kwon et al., 2023) for end2end model serving. Specifically, we add the zero-point support to its INT4 activation format, where originally the humming kernels only supported the symmetric version. Table 4 shows the end2end inference efficiency benchmark using an A100 GPU, where INT4 GEMM is run on the INT4 tensor cores. Compared to 16bit, OffQ is 1.5-1.7$\times$ the speed in prefill, and 2.0-2.4$\times$ the speed in decode. The per-group quantization overhead is negalectable in decode, while in prefill, group size 128 incurs an 27% to 48% overhead and group size 256 incurs 15% to 28% overhead compared to the 1-group per token setting (OffQ-g∞). Therefore, the group size is a tradeoff between accuracy and speed.

In terms of memory usage, the bit-width for weights is 4. For activations, the scale and zero-point storage overhead per group is negligible relative to the savings from 4-bit quantization. For example, with a group size of 128 and scale/zero-point stored in 16 bits each, the effective bit-width per activation is $^{(4\times128+16\times2)}/_{128} = 4.25$ bits, which is a substantial reduction from the original 16 bits.

6 Limitations and Future Work

While OffQ achieves strong performance under the W4A4KV4 setting, several limitations remain. First, we leave the integration of OffQ with complementary quantization techniques as future work, including non-uniform weight quantization (Savkin et al., 2025; Zandieh et al., 2026; Choi et al., 2025), more advanced weight quantization schemes (Li et al., 2025; Arai & Ichikawa, 2025; Zhang & Shrivastava, 2025; Zhang et al., 2026; Li et al., 2026), and rotation learning methods (Xiang & Zhang, 2025; Akhondzadeh et al., 2025; Shao et al., 2026). Second, we have not yet evaluated the method on additional low-precision data formats such as micro-scaling MXFP4 and NVFP4.

7 Conclusion

In this paper, we present OffQ, a post-training quantization method that addresses outlier-induced quantization error under the W4A4KV4 setting. The core of OffQ is an offsetting technique that converts outlier activation channels into a shared offset, which is then absorbed into the zero-point of asymmetric quantization. To identify the outlier subspace and concentrate the dominant outlier directions into as few channels as possible, we propose a Top-1 PCA method that captures the most significant activation variance efficiently. Extensive experiments on the Llama and Qwen 2.5 model families show that OffQ consistently outperforms state-of-the-art rotation-based PTQ methods across a wide range of model sizes and benchmarks, achieving lower perplexity and higher zero-shot accuracy while maintaining efficient memory usage. These results highlight the effectiveness of OffQ in enabling high-performance LLM inference on resource-constrained settings, paving the way for broader deployment of LLMs in real-world applications.

AI USE STATEMENT

In this work, we have not used generative AI tools for any of the tasks with required disclosure. We used generative AI tools for editing paper to improve readability, and for editing the kernel, vLLM integration, and speed benchmark codes. Note that the fake-quant implementation of OffQ, which is responsible for the perplexities/accuracies reported in experiments, was written manually without AI assistance. We have reviewed all AI-assisted work. We checked LLM-generated typo fixes, and LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Harshavardhan Adepu, Zhanpeng Zeng, Li Zhang, and Vikas Singh. Framequant: Flexible low-bit quantization for transformers. In *Proceedings of International Conference on Machine Learning (ICML)*, July 2024.

Mohammad Sadegh Akhondzadeh, Aleksandar Bojchevski, Evangelos Eleftheriou, and Martino Dazzi. KurTail : Kurtosis-based LLM quantization. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 17404–17419, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.943. URL https://aclanthology.org/2025.findings-emnlp.943/.

Yamato Arai and Yuma Ichikawa. Quantization error propagation: Revisiting layer-wise post-training quantization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=a3l3K9khbL.

Yamato Arai and Yuma Ichikawa. Quantization error propagation: Revisiting layer-wise post-training quantization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=a3l3K9khbL.

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. SliceGPT: Compress large language models by deleting rows and columns. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=vXxardq6db.

Saleh Ashkboos, Ilia Markov, Elias Frantar, Tingxuan Zhong, Xincheng Wang, Jie Ren, Torsten Hoefler, and Dan Alistarh. QUIK: Towards end-to-end 4-bit inference on generative large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3355–3371, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.197. URL https://aclanthology.org/2024.emnlp-main.197/.

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, 2024c.

Saleh Ashkboos, Mahdi Nikdan, Soroush Tabesh, Roberto L. Castro, Torsten Hoefler, and Dan Alistarh. HALO: Hadamard-assisted lower-precision optimization for LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=OcMpSh79aE.

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.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Understanding and overcoming the challenges of efficient transformer quantization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7947–7969, 2021.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=ayi7qezU87.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. Quip: 2-bit quantization of large language models with guarantees, 2024. URL https://arxiv.org/abs/2307.13304.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. EfficientQAT: Efficient quantization-aware training for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10081–10100, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.498. URL https://aclanthology.org/2025.acl-long.498/.

Euntae Choi, Sumin Song, Woosang Lim, and Sungjoo Yoo. Rotate, clip, and partition: Towards W2A4KV4 quantization by integrating rotation and learnable non-uniform quantizer. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 7568–7590, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.400. URL https://aclanthology.org/2025.findings-emnlp.400/.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

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, 2018. URL https://arxiv.org/abs/1803.05457.

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

Tim Dettmers, Ruslan A. Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. SpQR: A sparse-quantized representation for near-lossless LLM weight compression. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Q1u25ahSuy.

DaYou Du, Yijia Zhang, Shijie Cao, Jiaqi Guo, Ting Cao, Xiaowen Chu, and Ningyi Xu. BitDistiller: Unleashing the potential of sub-4-bit LLMs via self-distillation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 102–116, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.7. URL https://aclanthology.org/2024.acl-long.7/.

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. URL https://openreview.net/forum?id=tcbBPnfwxS.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Han Guo, Philip Greengard, Eric Xing, and Yoon Kim. LQ-loRA: Low-rank plus quantized matrix decomposition for efficient language model finetuning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=xw29VvOMmU.

Yefei He, Luoming Zhang, Weijia Wu, Jing Liu, Hong Zhou, and Bohan Zhuang. Zipcache: Accurate and efficient KV cache quantization with salient token identification. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=5t4ZAkPiJs.

Coleman Richard Charles Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=0LXotew9Du.

Chun Hu, Junhui He, Shangyu Wu, Yuxin He, Chun Jason Xue, and Qingan Li. Mlwq: Efficient small language model deployment via multi-level weight quantization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8078–8088, 2025a.

Xing Hu, Yuan Cheng, Dawei Yang, Zhixuan Chen, Zukang Xu, JiangyongYu, XUCHEN, Zhihang Yuan, Zhe jiang, and Sifan Zhou. OSTQuant: Refining large language model quantization with orthogonal and scaling transformations for better distribution fitting. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=rAcgDBdKnP.

Wei Huang, Yangdong Liu, Haotong Qin, Ying Li, Shiming Zhang, Xianglong Liu, Michele Magno, and Xiaojuan Qi. Billm: Pushing the limit of post-training quantization for llms. *arXiv preprint arXiv:2402.04291*, 2024.

Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and Xiaojuan Qi. SliM-LLM: Salience-driven mixed-precision quantization for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 25672–25692. PMLR, 13–19 Jul 2025.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2704–2713, 2018.

Minsoo Kim, Sihwa Lee, Suk-Jin Hong, Du-Seong Chang, and Jungwook Choi. Understanding and improving knowledge distillation for quantization aware training of large transformer encoders. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6713–6725, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.450. URL https://aclanthology.org/2022.emnlp-main.450/.

Sehoon Kim, Coleman Richard Charles Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. SqueezeLLM: Dense and sparse quantization, 2024a. URL https://openreview.net/forum?id=pZhdz4oyzo.

Sehoon Kim, Coleman Richard Charles Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. SqueezeLLM: Dense-and-sparse quantization. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 23901–23923. PMLR, 21–27 Jul 2024b. URL https://proceedings.mlr.press/v235/kim24f.html.

Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018.

Eldar Kurtic, Alexandre Noll Marques, Shubhra Pandit, Mark Kurtz, and Dan Alistarh. "give me BF16 or give me death"? accuracy-performance trade-offs in LLM quantization. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26872–26886, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1304. URL https://aclanthology.org/2025.acl-long.1304/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Jung Hyun Lee, Jeonghoon Kim, Se Jung Kwon, and Dongsoo Lee. Flexround: Learnable rounding based on element-wise division for post-training quantization. In *International Conference on Machine Learning*, pp. 18913–18939. PMLR, 2023.

Shuaiting Li, Juncan Deng, Kedong Xu, Rongtao Deng, Hong Gu, Minghan Jiang, Haibin Shen, and Kejie Huang. Rethinking residual errors in compensation-based LLM quantization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=LWYZ1nNkJl.

Yuhang Li, Ruokai Yin, Donghyun Lee, Shiting Xiao, and Priyadarshini Panda. GPTAQ: Efficient finetuning-free quantization for asymmetric calibration. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=QdELyl0FST.

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, 2024a.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for llm compression and acceleration. In *MLSys*, 2024b.

Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. Qserve: W4a8kv4 quantization and system co-design for efficient llm serving. *Proceedings of Machine Learning and Systems*, 7, 2025.

Jing Liu, Ruihao Gong, Xiuying Wei, Zhiwei Dong, Jianfei Cai, and Bohan Zhuang. QLLM: Accurate and efficient low-bitwidth quantization for large language models. In *International Conference on Learning Representations (ICLR)*, 2024a.

Yifei Liu, Jicheng Wen, Yang Wang, Shengyu Ye, Li Lyna Zhang, Ting Cao, Cheng Li, and Mao Yang. Vptq: Extreme low-bit vector post-training quantization for large language models. In *The 2024 Conference on Empirical Methods in Natural Language Processing*, 2024b.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. LLM-QAT: Data-free quantization aware training for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 467–484, Bangkok, Thailand, August 2024c. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.26. URL https://aclanthology.org/2024.findings-acl.26/.

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. URL https://openreview.net/forum?id=ogO6DGE6FZ.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. In *International Conference on Machine Learning*, pp. 32332–32344. PMLR, 2024d.

Yuexiao Ma, Huixia Li, Xiawu Zheng, Feng Ling, Xuefeng Xiao, Rui Wang, Shilei Wen, Fei Chao, and Rongrong Ji. Affinequant: Affine transformation quantization for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=of2rhALq8l.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL https://aclanthology.org/J93-2004/.

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

Meta AI. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/, September 2024. Accessed: 2026-05-05.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Lorenz K. Muller, Philippe Bich, Jiawei Zhuang, Ahmet Celik, Luca Benfenati, and Lukas Cavigelli. Sinq: Sinkhorn-normalized quantization for calibration-free low-precision llm weights, 2025. URL http://arxiv.org/abs/2509.22944.

James O'Neill and Sourav Dutta. Self-distilled quantization: Achieving high compression rates in transformer-based language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1329–1339, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.114. URL https://aclanthology.org/2023.acl-short.114/.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 1525–1534, 2016.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

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. *Journal of machine learning research*, 21(140):1–67, 2020.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4463–4473, 2019.

Semyon Savkin, Eitan Porat, Or Ordentlich, and Yury Polyanskiy. Nestquant: nested lattice quantization for matrix products and LLMs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=4OWGON33HE.

Utkarsh Saxena, Sayeh Sharify, Kaushik Roy, and Xin Wang. Resq: Mixed-precision quantization of large language models with low-rank residuals. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=4qIP1sXcR1.

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. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8Wuvhh0LYW.

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*, 2026. URL https://openreview.net/forum?id=LfcfwlLCHM.

Donghyun Son, Euntae Choi, and Sungjoo Yoo. NSNQuant: A double normalization approach for calibration-free low-bit vector quantization of KV cache. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=boNYskaXnO.

Yi Su, Yuechi Zhou, Quantong Qiu, Juntao Li, Qingrong Xia, Ping Li, Xinyu Duan, Zhefeng Wang, and Min Zhang. Accurate kv cache quantization with outlier tokens tracing. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12895–12915, 2025.

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

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, JiaxinHu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. Flatquant: Flatness matters for LLM quantization. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=uTz2Utym5n.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Vijay Thakkar, Pradeep Ramani, Cris Cecka, Aniket Shivam, Honghao Lu, Ethan Yan, Jack Kosaian, Mark Hoemmen, Haicheng Wu, Andrew Kerr, Matt Nicely, Duane Merrill, Dustyn Blasig, Aditya Atluri, Fengqi Qiao, Piotr Majcher, Paul Springer, Markus Hohnerbach, Jin Wang, and Manish Gupta. CUTLASS, January 2023. URL https://github.com/NVIDIA/cutlass.

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.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP$\#$: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *Forty-first International Conference on Machine Learning*, 2024a. URL https://openreview.net/forum?id=9BrydUVcoe.

Albert Tseng, Qingyao Sun, David Hou, and Christopher De. Qtip: Quantization with trellises and incoherence processing. *Advances in Neural Information Processing Systems*, 37:59597–59620, 2024b.

Mart van Baalen, Andrey Kuzmin, Markus Nagel, Peter Couperus, Cedric Bastoul, Eric Mahurin, Tijmen Blankevoort, and Paul Whatmough. Gptvq: The blessing of dimensionality in llm quantization. *arXiv preprint arXiv:2402.15319*, 2024.

vLLM Project. Humming. https://github.com/vllm-project/humming, 2026. High-performance JIT-compiled GEMM kernel library for quantized inference.

Haoqi Wang, Tong Zhang, and Mathieu Salzmann. Demystifying singular defects in large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=4yBnUokU2v.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. Bitnet: Scaling 1-bit transformers for large language models, 2023.

Xiuying Wei, Yunchen Zhang, Xiangguo Zhang, Ruihao Gong, Shanghang Zhang, Qi Zhang, Fengwei Yu, and Xianglong Liu. Outlier suppression: Pushing the limit of low-bit transformer language models. *Advances in Neural Information Processing Systems*, 35:17402–17414, 2022.

Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. Outlier suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1648–1665, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.102. URL https://aclanthology.org/2023.emnlp-main.102/.

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

Jingyang Xiang and Sai Qian Zhang. DFRot: Achieving outlier-free and massive activation-free for rotated LLMs with refined rotation. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=WzGypILLDb.

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 *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Yuzhuang Xu, Xu Han, Zonghan Yang, Shuo Wang, Qingfu Zhu, Zhiyuan Liu, Weidong Liu, and Wanxiang Che. Onebit: Towards extremely low-bit large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 66357–66382, 2024.

Yuzhuang Xu, Shiyu Ji, Qingfu Zhu, and Wanxiang Che. CRVQ: Channel-relaxed vector quantization for extreme compression of LLMs. *Transactions of the Association for Computational Linguistics (TACL)*, 13:1488–1506, 2025. URL https://doi.org/10.1162/TACL.a.45.

Haoqi Yang, Yao Yao, Zuchao Li, Baoyuan Qi, Liu Guoming, and Hai Zhao. XQuant: Achieving ultra-low bit KV cache quantization with cross-layer compression. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9785–9800, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.494. URL https://aclanthology.org/2025.emnlp-main.494/.

Zhihang Yuan, Lin Niu, Jiawei Liu, Wenyu Liu, Xinggang Wang, Yuzhang Shang, Guangyu Sun, Qiang Wu, Jiaxiang Wu, and Bingzhe Wu. Rptq: Reorder-based post-training quantization for large language models, 2023.

Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. Turboquant: Online vector quantization with near-optimal distortion rate. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=tO3ASKZlok.

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*, 2019.

Chao Zeng, Songwei Liu, Yusheng Xie, Hong Liu, Xiaojian Wang, Miao Wei, Shu Yang, Fangmin Chen, and Xing Mei. Abq-llm: Arbitrary-bit quantized inference acceleration for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22299–22307, 2025.

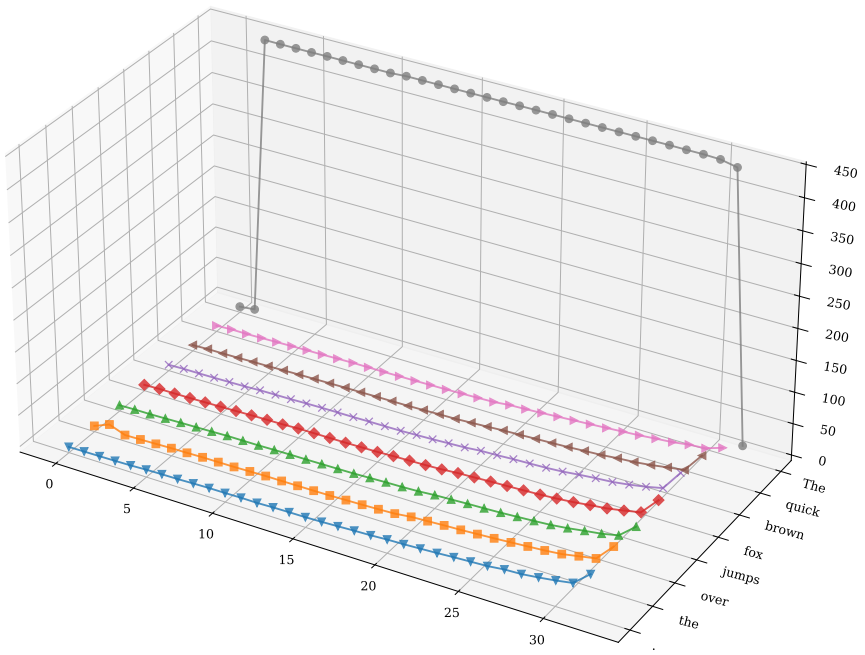
Aozhong Zhang, Naigang Wang, Yanxia Deng, Xin Li, Zi Yang, and Penghang Yin. Magr: Weight magnitude reduction for enhancing post-training quantization. *Advances in neural information processing systems*, 37:85109–85130, 2024a.

Shihao Zhang, Haoyu Zhang, Ian Colbert, and Rayan Saab. Qronos: Correcting the past by shaping the future... in post-training quantization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=7axclBCYul.

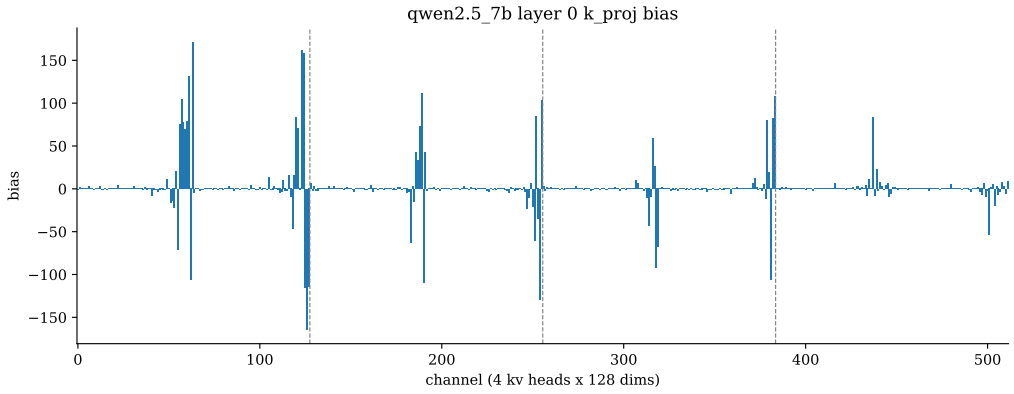
Tianyi Zhang and Anshumali Shrivastava. Leanquant: Accurate and scalable large language model quantization with loss-error-aware grid. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ISqx8giekS.

Yijia Zhang, Sicheng Zhang, Shijie Cao, Dayou Du, Jianyu Wei, Ting Cao, and Ningyi Xu. Afpq: Asymmetric floating point quantization for llms. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 28–36, 2024b.

Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, and Baris Kasikci. Atom: Low-bit quantization for efficient and accurate llm serving. In P. Gibbons, G. Pekhimenko, and C. De Sa (eds.), *Proceedings of Machine Learning and Systems*, volume 6, pp. 196–209, 2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/file/5edb57c05c81d04beb716ef1d542fe9e-Paper-Conference.pdf.



(a) Visualization of singular defects in Llama3-8B. The z-axis shows the norm of tokens across different layers. The y-axis corresponds to different input text tokens. The x-axis is the layer index, representing the output token after the i-th transformer layer, with the 0-th layer corresponding to the input embedding. The norm of the initial token "The" explodes at layer 2 and remains high until layer 31. This token is the singular defective token. An interesting property is that all singular defective tokens in a model share the same direction. Therefore, singular defects are a source of structured outliers. Top-1 PCA will pick the singular defective token from the input sequence via the L^∞ norm, and thus is accurate to extract the direction of this type of structural outlier.



(b) Visualization of the attention bias-induced outliers in Qwen2.5-7B. The bias of the key projection in the first layer of Qwen2.5-7B is plotted. The outlier channels contain values with magnitude greater than 100. Because the bias term is added to every token, it is a type of structured outlier.

Figure 4: Visualizing the two types of structured outliers in LLM activations.

A Visualization of Structured Outliers

We plot the two types of structured outliers in Figure 4. The first type, commonly referred to as singular defects in Wang et al. (2025) or massive activations in Sun et al. (2024), is a universal phenomenon in most contemporary LLMs. The second type is the bias term in the self-attention layers of certain LLMs, such as the Qwen 2.5 series.

Table 5: Comparison between OffQ and DuQuant. DuQuant uses W4A4, whereas OffQ uses W4A4KV4. The 0-Shot[6] is the average of PIQA, ARC-E, ARC-C, BoolQ, HellaSwag, WinoGrande.

		Llama3-8B		Llama3-70B		Llama2-7B		Llama2-13B	
Method	Setting	PPL	0-Shot[6]	PPL	0-Shot[6]	PPL	0-Shot[6]	PPL	0-Shot[6]
DuQuant	W4A4	8.06	67.72	4.90	76.60	6.08	61.01	5.33	63.42
OffQ	W4A4KV4	6.98	72.36	3.88	78.23	5.77	68.30	5.11	71.98

Table 6: Comparison between OffQ and FlatQuant using Llama3-8B. FlatQuant applies learnable clipping thresholds to activations, while OffQ here uses a fixed activation threshold of 0.95.

Method	Setting	PPL	ARC-C	ARC-E	HellaSwag	LAMBADA	PIQA	Winogrande	Average
FlatQuant	W4A4	6.90	50.51	75.88	76.49	73.20	79.00	72.93	71.33
	W4A4KV4	6.98	50.00	75.80	76.80	72.91	79.16	72.69	71.23
OffQ	W4A4	6.85	52.65	78.03	77.47	73.36	78.84	73.09	72.24
	W4A4KV4	6.97	51.45	77.69	77.00	73.45	78.78	71.90	71.71

B More Discussion on Design Choices

Why the L^∞ Norm. We assume a pre-normalization architecture with RMSNorm, consistent with most contemporary LLMs. Following QuaRot (Ashkboos et al., 2024c) and SpinQuant (Liu et al., 2025), we fuse the scaling parameters of RMSNorm into the adjacent weight matrices, so that the input activations to the linear layers of $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V$ in the self-attention module and $\boldsymbol{W}_U, \boldsymbol{W}_G$ in the feedforward module, lie on a scaled L^2 sphere. On a sphere, tokens with a few large-magnitude entries and many small entries exhibit a large L^∞ norm. For example, on the 3-dimensional unit sphere, the spiky token $[1, 0, 0]$ has a large L^∞ norm of 1, while the flat token $[0.58, 0.58, 0.58]$ has a smaller L^∞ norm of 0.58. The L^∞ norm therefore serves as a proxy for identifying the extreme outlier tokens in each sequence.

C Compare with More Baselines

DuQuant. The evaluation setup in DuQuant differs from ours in three ways: 1. DuQuant uses LWC (learnable weight clipping) instead of GPTQ. 2. DuQuant does not quantize the KV cache. 3. DuQuant reports zero-shot accuracy on only six tasks: PIQA, ARC-E, ARC-C, BoolQ, HellaSwag, and WinoGrande. Nevertheless, we compare DuQuant with OffQ in Table 5, and the results show that OffQ remains stronger even under the additional constraint of 4-bit KV cache.

FlatQuant. The evaluation setup of FlatQuant also differs from ours: 1. It focuses on W4A4 and reports W4A4KV4 only for Llama3-8B. 2. It evaluates zero-shot accuracy on a different set of six tasks: ARC-C, ARC-E, HellaSwag, LAMBADA (Paperno et al., 2016), PIQA, and WinoGrande. 3. It applies learnable clipping thresholds to activations. For a fair comparison, we apply a simple activation clipping threshold to OffQ in Table 6; this is not used in any other OffQ experiment reported in the paper. OffQ matches FlatQuant on perplexity (6.85 vs 6.90 at W4A4; 6.97 vs 6.98 at W4A4KV4) and remains ahead on average accuracy (72.24 vs 71.33; 71.71 vs 71.23), without relying on learned affine transforms.

D Detailed Results

We provide detailed quantization results for Llama and Qwen 2.5 models under W4A4KV4 quantization in Table 7 and Table 8, respectively.

Table 7: Detailed quantization results for Llama models under W4A4KV4 quantization. We report the perplexity (PPL ↓) on WikiText and the 0-shot accuracy across 8 tasks. Results for 16-bit, GPTQ, QUIK, QuaRot, SpinQuant, and ResQ are referenced from Saxena et al. (2025); and results for DFRot, KurTail, and OSTQuant are referenced from their respective papers.

	Method	**PPL**	ARC-c	ARC-e	BoolQ	HellaS	OBQA	PIQA	SIQA	WinoG	**Avg.**
Llama 3-8B	16-bit	6.1	53.2	77.1	81.1	79.2	44.8	80.9	47.0	73.4	67.09
	GPTQ	166.3	24.7	37.7	44.3	36.8	27.0	57.6	36.4	53.8	39.79
	QUIK	14.2	33.6	56.4	60.5	61.5	33.2	68.7	39.9	59.0	51.60
	QuaRot	7.8	45.1	70.4	73.8	74.7	42.6	76.6	45.1	68.5	62.10
	SpinQuant	7.4	48.0	75.4	75.8	75.4	**43.8**	77.5	45.0	69.2	63.76
	DFRot	7.91	44.97	71.09	73.27	74.13	43.00	78.24	44.58	69.53	62.35
	KurTail	7.2	48.2	75.4	79.2	76.4	43.6	78.4	45.8	70.0	64.63
	OSTQuant	7.29	49.32	76.73	78.87	76.01	43.20	78.51	45.70	69.22	64.70
	ResQ	7.1	49.2	75.0	72.5	76.5	43.0	78.3	45.8	**71.0**	63.91
	OffQ	**6.98**	**50.68**	**77.44**	**80.43**	**76.96**	**43.80**	**78.89**	**45.96**	69.77	**65.49**
Llama 3-70B	16-bit	2.9	64.2	85.9	85.3	84.9	48.6	84.4	50.8	80.6	73.09
	GPTQ	11655.0	25.9	26.0	37.9	26.2	28.6	50.4	34.3	49.9	34.90
	QUIK	8.0	44.5	68.9	60.7	75.0	36.4	76.1	43.2	60.4	58.15
	QuaRot	5.7	53.7	74.5	81.6	81.1	46.6	81.0	46.8	75.2	67.56
	SpinQuant	6.2	52.0	77.3	81.7	75.6	43.8	78.8	43.4	72.8	65.68
	DFRot	5.03	58.02	81.1	81.13	81.59	47.4	81.83	46.57	74.19	68.98
	KurTail	4.2	59.2	82.7	83.9	83.3	46.6	**83.5**	49.7	76.6	70.69
	OSTQuant	4.01	61.29	82.39	83.43	83.25	**48.93**	81.73	**51.24**	77.01	**71.16**
	ResQ	4.1	**61.4**	**84.3**	83.9	83.5	46.0	83.1	48.6	**78.3**	71.14
	OffQ	**3.88**	59.22	79.12	**86.12**	**84.21**	46.80	82.43	48.82	**78.30**	70.63
Llama 3-2-1B	16-bit	9.8	36.5	60.6	63.4	63.6	37.4	74.5	42.8	60.1	54.86
	GPTQ	108.9	24.7	32.7	52.3	30.7	23.6	54.3	34.4	51.1	37.98
	QUIK	21.8	27.4	46.0	55.0	46.0	26.4	62.4	38.6	52.6	44.30
	QuaRot	14.3	30.0	51.4	59.1	54.0	34.2	66.7	39.6	**57.1**	49.01
	SpinQuant	13.6	32.3	51.8	59.3	55.4	30.4	67.7	38.6	54.7	48.78
	KurTail	12.9	31.1	52.9	60.7	56.4	**36.4**	68.6	40.5	54.3	50.11
	ResQ	12.4	**34.0**	54.2	57.0	57.3	31.2	69.4	**41.0**	56.8	50.11
	OffQ	**12.32**	30.80	**55.22**	**62.26**	**57.55**	33.2	**70.95**	40.89	56.43	**50.91**
Llama 3-2-3B	16-bit	7.8	46.2	71.7	73.1	73.7	43.4	77.4	47.2	69.1	62.73
	GPTQ	178.3	27.0	27.0	48.8	44.4	27.8	59.1	37.1	51.5	40.34
	QUIK	15.8	32.9	50.1	52.6	59.1	33.2	68.7	40.3	53.0	48.74
	QuaRot	10.1	38.6	59.0	65.9	66.5	35.8	74.4	43.1	65.2	56.06
	SpinQuant	9.2	38.9	64.8	68.0	69.1	39.4	74.9	45.1	62.9	57.89
	KurTail	9.0	42.2	66.7	69.8	68.8	**39.8**	**75.6**	44.8	64.6	59.04
	ResQ	8.8	43.1	65.6	68.8	70.5	38.4	75.1	45.6	64.8	58.99
	OffQ	**8.78**	**44.8**	**70.41**	**71.62**	**71.49**	39.60	75.52	**45.65**	**67.32**	**60.80**
Llama 2-7B	16-bit	5.5	46.3	74.6	77.8	75.9	44.2	79.2	46.1	69.1	64.15
	GPTQ	9600	24.8	31.4	55.4	30.6	25.6	55.8	34.2	53.3	38.89
	QUIK	7.5	39.8	63.7	68.9	68.3	37.8	72.9	42.1	62.4	56.99
	QuaRot	6.1	41.5	71.4	73.2	73.2	40.6	76.9	43.6	65.6	60.75
	SpinQuant	6.0	43.6	71.3	73.8	73.2	40.4	76.0	44.1	65.4	60.98
	DFRot	6.25	43.52	70.83	73.3	72.62	41.40	76.82	44.17	65.11	60.97
	KurTail	5.9	43.1	72.0	72.0	73.2	41.2	76.6	45.6	66.8	61.31
	OSTQuant	5.91	42.92	72.56	74.71	73.14	**44.40**	77.42	**44.98**	66.77	**62.11**
	ResQ	5.8	**44.0**	**72.6**	**75.3**	**74.0**	41.0	77.9	43.9	66.9	61.95
	OffQ	**5.77**	43.77	70.79	74.89	73.84	41.80	**78.02**	44.27	**68.51**	61.99
Llama 2-13B	16-bit	4.9	49.1	77.4	80.5	79.4	45.2	80.7	47.2	72.1	66.45
	GPTQ	3120	23.6	31.1	38.7	27.2	26.8	53.6	35.8	49.8	35.83
	QUIK	6.8	43.7	68.0	71.3	73.3	40.0	75.7	45.1	64.6	60.21
	QuaRot	5.4	46.9	74.9	76.6	75.8	42.6	79.1	45.5	69.0	63.80
	SpinQuant	5.2	49.0	**76.3**	78.2	77.1	42.8	**79.3**	46.3	69.5	64.81
	DFRot	5.43	46.50	73.48	76.67	76.83	43.00	79.27	45.55	69.30	63.83
	KurTail	5.2	48.1	75.4	79.7	77.4	**45.0**	79.0	45.6	**71.2**	65.18
	OSTQuant	5.25	47.10	75.21	77.46	76.71	44.60	78.67	45.75	68.03	64.19
	ResQ	**5.1**	**49.1**	76.1	**79.7**	77.9	43.6	79.1	**46.6**	69.9	**65.25**
	OffQ	5.11	48.38	**76.30**	79.36	**77.96**	43.6	79.05	46.52	70.8	**65.25**

Table 8: Detailed quantization results for Qwen 2.5 models under W4A4KV4 quantization. We report the perplexity (PPL ↓) on WikiText and the 0-shot accuracy across 8 tasks. Results for 16-bit, GPTQ, QUIK, QuaRot, and ResQ are referenced from Saxena et al. (2025).

	Method	**PPL**	ARC-c	ARC-e	BoolQ	HellaS	OBQA	PIQA	SIQA	WinoG	**Avg.**
Qwen 2.5-1.5B	16-bit	9.3	45.1	72.1	72.9	67.7	40.2	76.3	48.8	63.7	60.85
	GPTQ	25770	23.9	26.9	43.9	26.1	27.6	49.7	32.1	51.5	35.21
	QUIK	6614	21.8	31.9	40.9	27.9	27.4	52.8	35.2	48.6	35.81
	QuaRot	6600	23.6	37.3	46.2	28.6	27.0	56.3	35.2	52.4	38.33
	ResQ	12.5	38.7	64.1	65.7	61.4	**37.8**	71.6	42.7	**60.1**	55.26
	OffQ	**11.35**	**40.87**	**69.74**	**70.73**	**63.27**	**37.80**	**73.12**	**44.93**	59.75	**57.53**
Qwen 2.5-3B	16-bit	8.0	47.4	73.0	77.5	73.6	42.0	78.7	49.9	68.4	63.81
	GPTQ	9978	26.0	26.7	41.5	26.7	28.2	51.5	31.9	48.3	35.10
	QUIK	15.5	36.1	55.4	61.4	57.2	36.2	67.1	40.8	55.3	51.19
	QuaRot	68.8	32.4	53.1	51.6	49.2	33.4	66.7	39.3	56.4	47.76
	ResQ	9.0	45.3	70.5	**72.7**	**70.2**	**42.4**	76.8	46.7	**64.4**	61.13
	OffQ	**8.98**	**47.10**	**73.90**	70.76	69.94	40.60	**77.15**	**48.31**	64.01	**61.47**
Qwen 2.5-7B	16-bit	6.8	51.2	77.6	84.7	78.9	47.2	80.0	54.8	73.2	68.45
	GPTQ	13594	25.2	25.6	37.8	26.3	28.2	52.4	34.4	48.9	34.85
	QUIK	260.3	29.5	42.4	51.7	36.3	28.2	59.6	34.5	49.6	41.48
	QuaRot	4036	25.9	41.0	39.1	29.1	27.6	57.9	35.7	50.6	38.36
	ResQ	8.2	49.0	74.7	81.4	75.7	**45.0**	**78.9**	49.4	68.2	65.29
	OffQ	**7.66**	**49.23**	**74.96**	**83.30**	**76.49**	44.80	78.56	**52.71**	**69.22**	**66.16**
Qwen 2.5-14B	16-bit	5.3	58.8	79.4	85.4	82.9	45.4	81.9	55.3	75.8	70.61
	GPTQ	5100	23.8	29.1	47.7	30.1	27.6	51.3	34.6	51.2	36.93
	QUIK	10.5	45.0	67.1	64.7	68.9	37.6	74.8	43.9	59.3	57.66
	QuaRot	6.8	54.8	79.6	79.9	78.7	44.0	79.5	49.9	70.7	67.14
	ResQ	6.2	**57.6**	**82.1**	84.9	**81.1**	**44.8**	80.5	51.7	70.6	69.16
	OffQ	**6.07**	56.40	80.26	**85.17**	81.00	43.60	**80.79**	**52.46**	**73.95**	**69.20**
Qwen 2.5-32B	16-bit	5.0	55.7	78.0	87.4	84.1	44.4	82.3	56.4	75.2	70.44
	GPTQ	3891	25.4	35.4	48.5	31.8	27.0	53.8	35.8	50.5	38.53
	QUIK	9.6	41.0	64.6	74.9	72.0	39.6	75.8	44.5	60.2	59.08
	QuaRot	6.1	54.5	76.1	85.1	81.5	44.2	80.1	51.3	70.4	67.90
	ResQ	5.6	55.1	**78.4**	86.0	82.5	**45.4**	81.1	**53.9**	74.0	69.55
	OffQ	**5.52**	**55.12**	78.07	**87.03**	**82.58**	45.00	**81.94**	52.05	**74.90**	**69.59**
Qwen 2.5-72B	16-bit	3.9	62.6	83.2	89.2	86.0	46.6	83.6	58.4	77.7	73.41
	GPTQ	37967	25.4	25.8	38.1	25.6	26.6	51.2	34.2	49.4	34.54
	QUIK	8.3	45.1	68.1	77.2	77.2	39.0	77.4	45.6	65.6	61.90
	QuaRot	4.9	55.8	81.1	87.5	84.0	45.2	81.7	52.5	74.5	70.28
	ResQ	4.6	58.4	80.9	88.4	84.9	**48.2**	82.6	55.5	**77.0**	71.98
	OffQ	**4.29**	**61.95**	**81.99**	**88.93**	**85.28**	46.80	**83.46**	**56.50**	76.56	**72.68**