DFRot: Achieving Outlier-Free and Massive Activation-Free for Rotated LLMs with Refined Rotation

Jingyang Xiang
New York University
xiangxiangjingyang@gmail.com

Sai Qian Zhang
New York University
sai.zhang@nyu.edu

Abstract

Rotating the activation and weight matrices to reduce the influence of outliers in large language models (LLMs) has recently attracted significant attention, particularly in the context of model quantization. Prior studies have shown that in low-precision quantization scenarios, such as 4-bit weights and 4-bit activations (W4A4), randomized Hadamard transforms can achieve significantly higher accuracy than randomized orthogonal transforms. Notably, the reason behind this phenomenon remains unknown. In this paper, we find that these transformations show substantial improvement in eliminating outliers for common tokens and achieve similar quantization error. The primary reason for the accuracy difference lies in the fact that randomized Hadamard transforms can slightly reduce the quantization error for tokens with massive activations while randomized orthogonal transforms increase the quantization error. Due to the extreme rarity of these tokens and their critical impact on model accuracy, we consider this a long-tail optimization problem, and therefore construct a simple yet effective method: a weighted loss function. Additionally, we propose an optimization strategy for the rotation matrix that involves alternating optimization of quantization parameters while employing orthogonal Procrustes transforms to refine the rotation matrix. This makes the distribution of the rotated activation values more conducive to quantization, especially for tokens with massive activations. Our method enhances the Rotated LLMs by achieving dual free, *Outlier-Free* and *Massive Activation-Free*, dubbed as *DFRot*. Extensive experiments demonstrate the effectiveness and efficiency of DFRot. By tuning the rotation matrix using just a single sample, DFRot achieves a perplexity improvement of 0.98 and 0.95 on W4A4KV4 and W4A4KV16, respectively, for LLaMA3-70B, a model known for its quantization challenges. Code is available at https://github.com/JingyangXiang/DFRot.

1 Introduction

Large Language Models (LLMs) have shown exceptional abilities across numerous domains. Cutting-edge open-source models like LLaMA (Touvron et al., 2023) and Mistral (Jiang et al., 2023), along with proprietary LLMs such as GPT (Achiam et al., 2023) and Gemini (Team et al., 2023), are now being applied in a wide range of applications, including natural language understanding (Zellers et al., 2019; Hendrycks et al., 2020), machine translation (Zhang et al., 2023), content generation (Mo et al., 2024), recommendation systems (Wu et al., 2023; Wang et al., 2024; 2025) and agent (Li et al., 2025).

However, the remarkable success of LLMs is largely reliant on significant computational resources. LLMs often consist of billions of parameters, making them not only resource-intensive to train but also challenging to deploy on devices with limited computational capacity, such as mobile phones and edge devices. Additionally, the high memory and processing demands not only drive up hardware costs but also significantly increase energy consumption, leading to serious deployment concerns. To address these challenges, researchers and engineers are actively exploring various model compression techniques (Fran-

tar et al., 2022; Xiao et al., 2023; Lin et al., 2024a; Yao et al., 2022; Frantar & Alistarh, 2023; Ashkboos et al., 2024a; Wei et al., 2024; Zhao et al., 2025). These techniques aim to reduce the size of LLMs while maintaining their performance as effectively as possible, achieving a balance between efficiency and accuracy.

Unfortunately, the presence of outliers in the activations (Dettmers et al., 2022; Zeng et al., 2022) often leads to a significant reduction in model accuracy when PTQ is applied directly. To address this problem, earlier approaches have either scaled weights and activations (Xiao et al., 2023; Wei et al., 2023; Shao et al., 2023), shifting the quantization challenges from activations to weights, or employed mixed-precision techniques to isolate outliers (Dettmers et al., 2022), thereby minimizing the LLM's quantization error.

Recent research (Ashkboos et al., 2024b) has demonstrated that rotating activations in LLMs can effectively eliminate most outliers while preserving computational invariance, ensuring that the LLM's output remains identical to its original results. Moreover, the rotation matrices can be merged into the weights, imposing no additional burden on network inference. This innovative computational invariance (Ashkboos et al., 2024a) has garnered significant attention from researchers.

Although rotation is widely recognized as an important method for the quantization of LLMs, there remain many unresolved issues. For example, as shown in Table 1, when activations are reduced to 4-bit, the reasons why randomized Hadamard transforms (RH) often achieve significant improvement compared to randomized orthogonal transforms (RO) (Ashkboos et al., 2024b; Liu et al., 2024) have not yet been fully understood. However, while directly training rotation matrices can yield good results (Liu et al., 2024), the training process will cause substantial computational resources and adds complexity to the quantization process.

In this paper, we first investigate the underlying reasons why RH outperforms RO. We find that for ordinary tokens consisting primarily of outliers (Achiam et al., 2023), both RO and RH transformations can equally reduce quantization error when applied to these tokens. As shown in Figure 3, in terms of quantization error, there is no substantial difference between the two transformations. In contrast, for special tokens with *massive activations* (Sun et al., 2024), using RO on these activations surprisingly leads to an increase in quantization error. Our experiments show that this inability to efficiently manage massive activations greatly restricts the accuracy of quantized LLMs. On the other hand, while RH performs better than RO, it only manages to maintain or slightly reduce the quantization error for these large activations. This observation indicates that both transformation methods struggle to effectively manage massive activations in LLM quantization.

Building on these insights, we propose a novel optimization method to enhance the performance of quantized LLMs, achieving both *Outlier-Free* and *Massive Activation-Free*, *e.g.* dual free (*DFRot*). By treating scarce tokens with massive activations as long-tail distributed data, we develop a simple yet effective weighted loss function. Additionally, we introduce an alternating optimization approach to refine the rotation matrices and quantization parameters, further minimizing quantization error. Extensive experiments demonstrate the effectiveness of our proposed method. Specifically, by tuning the rotation matrix with just a single sample, DFRot achieves a PPL improvement of 0.95 and 0.98 on W4A4KV4 and W4A4KV16 for LLaMA3-70B with WikiText-2, a model recognized for its quantization challenges (Huang et al., 2024).

2 Related Work

2.1 Eliminating outliers via Scale Invariance

The initial idea behind suppressing outliers through scale invariance stems from the observation that weights are easier to quantize than activations, and outliers in activations often appear in a few fixed channels Dettmers et al., 2022. Based on this, SmoothQuant (Xiao et al., 2023) first proposes that we can offline migrate the quantization difficulty from activations to weights via scale invariance. SmoothQuant enables an INT8 quantization of both

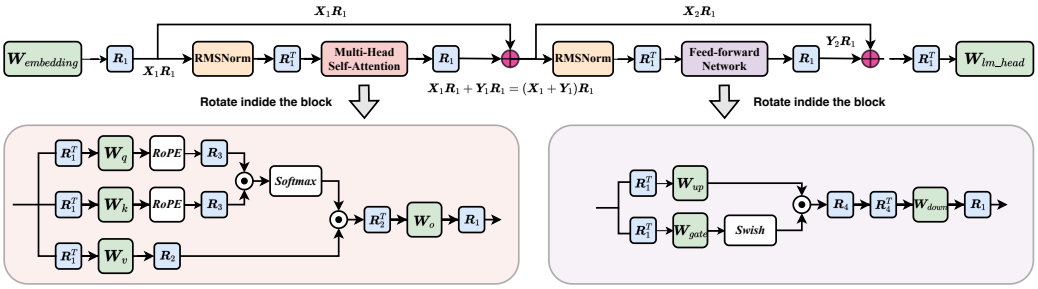


Figure 1: An illustration of rotational invariance in the LLaMA architecture. The rotation matrix $\boldsymbol{R}_1$ can be integrated into the residual connection, ensuring the network retains rotational invariance. The rotation inner the block can further reducing outliers within block. Both of them make LLM fewer outliers and be easier to quantize. The rotation matrix $\boldsymbol{R}_1$, $\boldsymbol{R}_1^T$, $\boldsymbol{R}_2$, $\boldsymbol{R}_2^T$ and $\boldsymbol{R}_4^T$ can be integrated into weights. $\boldsymbol{R}_3$ and $\boldsymbol{R}_4$ need to compute online.

weights and activations for all the matrix multiplications in LLMs. Furthermore, Outlier Suppression+ (Wei et al., 2023) proposes a fast and stable scheme to effectively calculate scaling values, achieving a better balance in quantization burden. To reduce manual design and further enhance quantization performance in extremely low-bit quantization, OmniQuant (Shao et al., 2023) introduces Learnable Weight Clipping and Learnable Equivalent Transformation, efficiently optimizing the quantization process for both weight-only and weight-activation quantization. In the clipping W4A8 quantization, QQQ (Zhang et al., 2024) proposes to dynamically handle outliers through adaptive smoothing. QServe (Lin et al., 2024b) proposes SmoothAttention to effectively mitigate the accuracy degradation caused by 4-bit KV quantization. Both QQQ and QServe have effectively enhanced the performance of LLMs in W4A8 quantization.

2.2 Eliminating outliers via Rotational Invariance

Although scale invariance can reduce outliers and improve quantization performance, it merely transfers the outliers from activations to weights and has not eliminated them fundamentally. When the magnitude of the outliers is large, scaling struggles to achieve an effective balance between weights and activations. Recently, researchers have found that applying rotation matrices to networks can effectively reduce outliers without increasing the complexity of LLMs. QuIP Chee et al. (2024) is the first to suggest that quantization can benefit from the incoherence between weight and Hessian matrices. It employed randomized orthogonal matrices generated by Kronecker product to enhance their incoherence. QuIP# (Tseng et al., 2024) replaces the randomized orthogonal matrices with randomized Hadamard matrices, which are faster and possess better theoretical properties. QuaRot (Ashkboos et al., 2024b) is the first work to apply rotational invariance (Ashkboos et al., 2024a) for model quantization. QuaRot finds that randomized Hadamard transformations yield better results compared to randomized orthogonal transformations. SpinQuant (Liu et al., 2024) and OSTQuant (Hu et al., 2025) further extends the rotation matrices to a trainable space and applied Cayley optimization (Li et al., 2020) to refine them, achieving significant improvements across diverse datasets.

3 Rotational Invariance, Quantization and Massive Activation

3.1 Rotational Invariance

First, we briefly introduce rotational invariance in LLMs, using the structure of LLaMA as an example. We assume that the α in the RMSNorm has been fused into the follow linear layers' weights, including $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, $\boldsymbol{W}_v$, $\boldsymbol{W}_{up}$ and $\boldsymbol{W}_{gate}$ and RMSNorm applies to each row of the activations $\boldsymbol{X}$ as $\boldsymbol{X}_{i,:} \leftarrow \boldsymbol{X}_{i,:} / \|\boldsymbol{X}_{i,:}\|$. If $\boldsymbol{R}_1$ is an rotation matrix, we have the commutation property $\text{RMSNorm}(\boldsymbol{X}\boldsymbol{R}_1) = \text{RMSNorm}(\boldsymbol{X})\boldsymbol{R}_1$ (Ashkboos et al., 2024a). This property implies that multiplying the input of RMSNorm by $\boldsymbol{R}_1$ is equivalent to multiplying the RMSNorm output by $\boldsymbol{R}_1$.

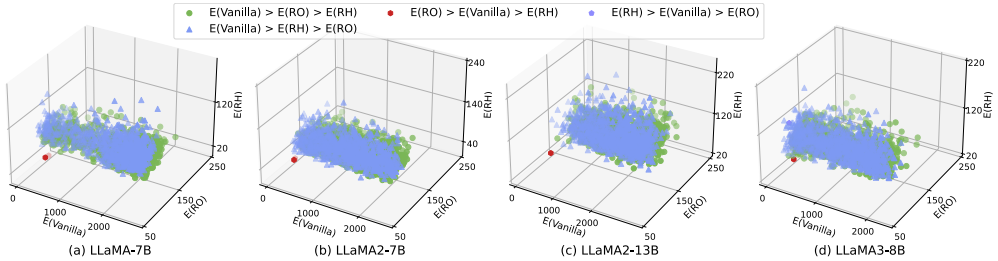


Figure 2: Comparison of 4-bit activation quantization error $\mathrm{E}(\cdot)$ for each token with NR, RO and RH for (a) LLaMA-7B, (b) LLaMA2-7B, (c) LLaMA2-13B and (d) LLaMA3-8B. The tokens are from model.layers.6.post_attention_layernorm. Best viewed in color.

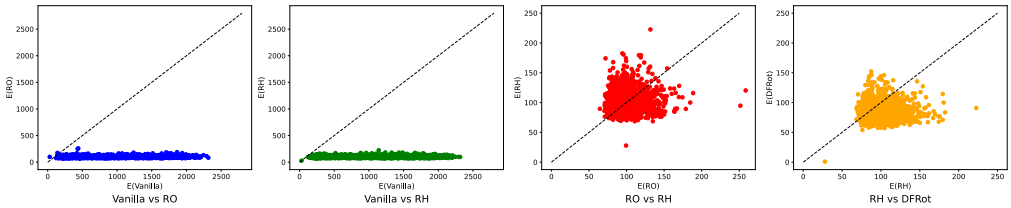


Figure 3: Comparison of 2D 4-bit quantization errors for tokens with NR, RO, RH and DFRot for LLaMA3-8B from Figure 2.

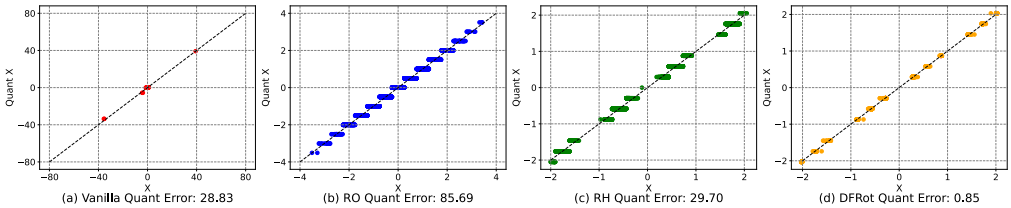


Figure 4: Comparison of 4-bit quantization error for the token with massive activation with NR, RO, RH and DFRot for LLaMA3-8B from Figure 2.

As shown in Figure 1, to remove outliers in the input activations, a rotation matrix $\boldsymbol{R}_1$ is applied to the embedding layer $\boldsymbol{W}_{embedding}$, resulting in a new input activation $\boldsymbol{X}_1\boldsymbol{R}_1$. According to the above, we can know once we transform $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, $\boldsymbol{W}_v$ and $\boldsymbol{W}_o$ in the Multi-Head Attention (MHA) to $\boldsymbol{R}_1^T\boldsymbol{W}_q$, $\boldsymbol{R}_1^T\boldsymbol{W}_k$, $\boldsymbol{R}_1^T\boldsymbol{W}_v$ and $\boldsymbol{W}_o\boldsymbol{R}_1$, the hidden feature within the MHA will remain unchanged, and the original output feature $\boldsymbol{Y}_1$ will become $\boldsymbol{Y}_1\boldsymbol{R}_1$. The following Feed-Forward Network's input $\boldsymbol{X}_2$ from the residual connection will be modified to $(\boldsymbol{X}_1 + \boldsymbol{Y}_1)\boldsymbol{R}_1 = \boldsymbol{X}_2\boldsymbol{R}_1$. If we further transform $\boldsymbol{W}_{up}$, $\boldsymbol{W}_{gate}$ and $\boldsymbol{W}_{down}$ to $\boldsymbol{R}_1^T\boldsymbol{W}_{up}$, $\boldsymbol{R}_1^T\boldsymbol{W}_{gate}$ and $\boldsymbol{W}_{down}\boldsymbol{R}_1$, the hidden feature within the FFN will also remain unchanged, and the output feature $\boldsymbol{X}_3$ will be modified to $(\boldsymbol{X}_2 + \boldsymbol{Y}_2)\boldsymbol{R}_1 = \boldsymbol{X}_3\boldsymbol{R}_1$. Based on mathematical induction, we can get that $\boldsymbol{X}_n\boldsymbol{R}_1 + \boldsymbol{Y}_n\boldsymbol{R}_1 = (\boldsymbol{X}_n + \boldsymbol{Y}_n)\boldsymbol{R}_1 = \boldsymbol{X}_{n+1}\boldsymbol{R}_1$ for the n-th module. To this end, by transforming $\boldsymbol{W}_{lm\_head}$ into $\boldsymbol{R}_1^T\boldsymbol{W}_{lm\_head}$, the network output will remain unchanged.

There is also rotational invariance within the block. For MHA, we can insert head-wise rotation matrices $\boldsymbol{R}_2$ and $\boldsymbol{R}_2^T$ for $\boldsymbol{W}_v$ and $\boldsymbol{W}_o$ and $\boldsymbol{R}_3$ for **Query** and **Key** after RoPE. For FFN, we can insert $\boldsymbol{R}_4$ and $\boldsymbol{R}_4^T$ between Swish and $\boldsymbol{W}_{down}$. These approaches can further eliminate outliers and reduce quantization error while keeping the block output unchanged. In this paper, we only discuss $\boldsymbol{R}_1$. For $\boldsymbol{R}_2$, $\boldsymbol{R}_3$, and $\boldsymbol{R}_4$, we follow the QuaRot (Ashkboos et al., 2024b) settings and use Hadamard matrices.

3.2 Why the Randomized Hadamard is better than Randomized Orthogonal?

Based on the computational invariance described in Section 3.1, it is evident that the choice of rotation matrices is critical for ensuring the accuracy performance of the quantized model. Therefore, a natural question arises: **What type of rotation matrix offers the most advantageous properties?**

Method	LLaMA-7B			LLaMA2-7B			LLaMA2-13B			LLaMA3-8B		
	4-4-4	4-4-16	4-8-16	4-4-4	4-4-16	4-8-16	4-4-4	4-4-16	4-8-16	4-4-4	4-4-16	4-8-16
GPTQ	NaN	NaN	NaN	NaN	NaN	NaN	Inf	Inf	6.01	Inf	Inf	7.29
(RO) QuaRot	6.68	6.62	5.80	7.96	7.71	5.61	6.00	5.92	4.99	10.54	10.15	6.52
(RO) QuaRot.FP16()	**6.30**	**6.27**	-	**6.17**	**6.10**	-	**5.38**	**5.34**	-	**7.83**	**7.68**	-
(RH) QuaRot	6.37	6.33	5.81	6.27	6.20	5.61	5.51	5.46	5.01	8.20	8.02	6.52
(RH) QuaRot.FP16()	**6.30**	**6.28**	-	**6.17**	**6.10**	-	**5.40**	**5.37**	-	**7.82**	**7.67**	-

Table 1: WikiText-2 perplexity (↓) results for RO and RH for LLaMA models. The 4-4-4, 4-4-16, 4-8-16 represent W4A4KV4, W4A4KV16, W4A8KV16 respectively. We show the failed GPTQ using NaN and the perplexity results>100 by Inf. QuaRot.FP16() denotes retaining tokens with massive activations as FP16.

We begin by focusing on RO and RH, as both QuaRot (Ashkboos et al., 2024b) and SpinQuant (Liu et al., 2024) have demonstrate that RH delivers substantial improvements over RO in LLMs. We conducted experiments by applying RO and RH to the LLaMA models respectively, followed by weight quantization using GPTQ under various quantization settings. The results are shown in Table 1. Benefiting from the outlier elimination through rotational invariance, we find that for dynamical token-wise 8-bit activation quantization, both RO and RH lead to significant performance improvements compared to standard quantization. Additionally, no substantial performance difference is observed between the two transformations. **However, under 4-bit dynamical token-wise activation quantization,** RH **significantly outperforms** RO.

To investigate the performance differences between RH and RO under 4-bit activation setting, we plot the corresponding quantization error after applying 4-bit quantization to the multiple tokens. We also display the quantization error for the baseline setting where quantization is applied without rotating the activation to better understand the impact of using the rotation matrix. As shown in Figure 2, compared to the no rotation (NR), both RO and RH effectively reduce the quantization error for most tokens across different models. While RH slightly lowers the quantization error, the difference between the two methods is minimal for the majority of tokens. This leads to the question: **What explains the significant difference in PPL during quantization when their quantization errors are so similar?**

To answer this question, we turn our attention to massive activation (Sun et al., 2024), a rare but significant feature in LLMs. As shown in Figure 2, the red points represent quantization error for the tokens with massive activation. While most tokens show large quantization errors under NR, these special tokens display significantly smaller errors, which can be observed from Figure 3. It is normal since each token has a fixed L_2 norm after RMSNorm processing, as shown in Figure 4(a), tokens with massive activation naturally exhibit smaller quantization errors when quantized to 4-bit. Figure 4 presents the quantization result for the token with massive activation after applying NR, RO, and RH. Surprisingly, the rotation operations do not significantly reduce quantization errors for these tokens. In fact, compared to NR, RO greatly increases their quantization error, while RH only marginally reduces it. **This leads us to question whether tokens with massive activation are the primary cause of the significant accuracy discrepancies between** RH **and** RO**.**

To investigate this further, we build upon QuaRot by retaining tokens with massive activations in FP16 format for both RO and RH, while applying 4-bit quantization to the remaining input tokens, denoted as (RO) QuaRot.FP16() and (RH) QuaRot.FP16(). As shown in Table 1, for all LLaMA models, the performance gap between RH QuaRot and RO QuaRot is totally disappeared. It is so surprising that by simply retaining these extremely few tokens (often less than one-thousandth) as FP16, we can completely eliminate the performance difference between RO and RH. Therefore, we can make the following conclusion:

Why the Randomized Hadamard is better than Randomized Orthogonal?

RH = RO + Tokens with Massive Activations: RH is better than RO because it performs more effectively when reducing the quantization error for tokens with massive activations in 4-bit activation quantization.

3.3 Optimization Objectives and Calibration Data Selection

As mentioned above, although retaining tokens with massive activations as high-precision floating-point numbers can significantly enhance model accuracy, this approach is akin to a token-level version of LLM.int8(). It still requires fine-grained mixed-precision computations during the process, which will introduce additional system level optimization. Therefore, in this paper, we focus on W4A4 quantization to maintain simplicity and efficiency in the computation process. We consider a loss function of the following form:

$$\mathcal{L}(\boldsymbol{R}_1, \boldsymbol{g}_{\boldsymbol{x}}) = \mathbb{E}_{\boldsymbol{x}} \left[\|\boldsymbol{x}\boldsymbol{R}_1 - \mathcal{Q}(\boldsymbol{x}\boldsymbol{R}_1, \boldsymbol{g}_{\boldsymbol{x}})\|_2^2 \right], \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^{1\times C}$ is the token vector from a calibration dataset $\boldsymbol{X}^{cal} \in \mathbb{R}^{L\times C}$. C is the hidden size and L is the number of tokens. $\boldsymbol{R}_1 \in \mathbb{R}^{C\times C}$ satisfies $\boldsymbol{R}_1\boldsymbol{R}_1^T = \boldsymbol{I}$, $\boldsymbol{g}_{\boldsymbol{x}}$ is the quantization parameters and $\mathcal{Q}(\boldsymbol{x}, \boldsymbol{g}_{\boldsymbol{x}}) \in \mathbb{R}^{1\times C}$ is the quantization of the $\boldsymbol{x}$. The expectation $\mathbb{E}[\cdot]$ is taken over the token distribution. For the ease of analysis, we use the mean squared error $\|\cdot\|_2$.

Meanwhile, we introduce our data selection principle. We denote the calibration dataset as $\boldsymbol{X}$, the tokens with massive activations as $\boldsymbol{X}^m$, and the remaining tokens as $\boldsymbol{X} \setminus \boldsymbol{X}^m$:

$$\mathcal{L}(\boldsymbol{R}_1, \boldsymbol{g}_{\boldsymbol{x}}) = \mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}^{cal}\setminus\boldsymbol{X}^m} \left[\|\boldsymbol{x}\boldsymbol{R}_1 - \mathcal{Q}(\boldsymbol{x}\boldsymbol{R}_1, \boldsymbol{g}_{\boldsymbol{x}})\|_2^2 \right] + \gamma^2 \mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}^m} \left[\|\boldsymbol{x}\boldsymbol{R}_1 - \mathcal{Q}(\boldsymbol{x}\boldsymbol{R}_1, \boldsymbol{g}_{\boldsymbol{x}})\|_2^2 \right]. \tag{2}$$

During calibration, we apply a weighted loss to prioritize the quantization error on tokens with massive activations, with γ representing the weight.

The motivation behind this principle stems from the observations in Table 1. Since $\boldsymbol{X}^m$ is the key factor contributing to the performance gap between RO and RH, simply optimizing $\boldsymbol{R}_1$ via Eq. 1 fails to specifically target $\boldsymbol{X}^m$. On the other hand, compared to the NR in Table 1, RO also significantly improves performance, indicating that reducing the outliers on $\boldsymbol{X}^{cal} \setminus \boldsymbol{X}^m$ can enhance quantization performance, optimizing only for $\boldsymbol{X}^m$ has the risk of increasing the quantization error for $\boldsymbol{X}^{cal} \setminus \boldsymbol{X}^m$, ultimately degrading the model's performance. Hence, it is crucial to optimize both $\boldsymbol{X}^m$ and $\boldsymbol{X}^{cal} \setminus \boldsymbol{X}^m$. Naturally, we can regard this a long-tail optimization problem, where $\boldsymbol{X}^m$ represents the long-tail but important data. Using a weighted approach to optimize the quantization loss is a simple yet highly effective method. Ablation studies in Section 4.2 further demonstrate the advantages of this strategy.

3.4 Solution Methods

Optimizing $\boldsymbol{R}_1$ is a challenging task. Since $\boldsymbol{R}_1$ influences every MHA and FFN in the network, adjusting the activation distribution in one layer impacts the quantization results across all layers. This makes it difficult to optimize layer by layer or block by block (Shao et al., 2023; Wei et al., 2023). A straightforward approach is to use training methods for quantization-aware fine-tuning of the rotation matrix across the entire network (Liu et al., 2024). Although it does not require retaining the gradients of the weights or the corresponding optimizer states, it still demands substantial computational resources during the quantization process.

In this paper, we focus on improving the effectiveness of rotation matrices in mitigating outliers and massive activation. Intuitively, we hypothesize that a rotation matrix that minimizes quantization error will lead to better performance. Drawing inspiration from Simsiam (Chen & He, 2021), we propose to regard quantization representation $\mathcal{Q}(\boldsymbol{x}\boldsymbol{R}_1, \boldsymbol{g})$ as cluster centroids $\boldsymbol{\eta}_{\boldsymbol{x}}$. In the context, optimizing $\boldsymbol{R}_1$ and $\boldsymbol{g}$ is equivalent to optimizing $\boldsymbol{R}_1$ and $\boldsymbol{\eta}_{\boldsymbol{x}}$, which can be viewed as an implementation of an Expectation-Maximization (EM)-like algorithm, as shown in the following equation:

$$\begin{aligned} \min_{\boldsymbol{R}_1, \boldsymbol{\eta}_{\boldsymbol{x}}} \mathcal{L}(\boldsymbol{R}_1, \boldsymbol{\eta}_{\boldsymbol{x}}) &= \mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}^{cal}\setminus\boldsymbol{X}^m} \left[\|\boldsymbol{x}\boldsymbol{R}_1 - \boldsymbol{\eta}_{\boldsymbol{x}}\|_2^2 \right] + \gamma^2 \mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}^{cal}} \left[\|\boldsymbol{x}\boldsymbol{R}_1 - \boldsymbol{\eta}_{\boldsymbol{x}}\|_2^2 \right] \\ &= \mathbb{E}_{\boldsymbol{x}\in\widehat{\boldsymbol{X}}^{cal}} \left[\|\boldsymbol{x}\boldsymbol{R}_1 - \boldsymbol{\eta}_{\boldsymbol{x}}\|_2^2 \right], \end{aligned} \tag{3}$$

where $\boldsymbol{\eta}_{\boldsymbol{x}} = \mathcal{Q}(\boldsymbol{x}\boldsymbol{R}_1, \boldsymbol{g})$ and $\widehat{\boldsymbol{X}}^{cal} = \{\boldsymbol{x} | \boldsymbol{x} \in \boldsymbol{x} \in \boldsymbol{X}^{cal} \setminus \boldsymbol{X}^m\} \cup \{\gamma\boldsymbol{x} | \boldsymbol{x} \in \boldsymbol{X}^m\}$. This formulation is analogous to k-means clustering (Macqueen, 1967), and $\boldsymbol{R}_1$ and $\boldsymbol{\eta}_{\boldsymbol{x}}$ act like the kernel

function and cluster centroids, respectively. Similar to k-means clustering, the problem described in Eq 3 can be approached using an alternating algorithm, where one set of variables is fixed while solving for the other. Formally, we can alternate between solving these two subproblems:

$$\boldsymbol{\eta}_{\boldsymbol{x}}^{t} \leftarrow \arg\min_{\boldsymbol{\eta}_x} \mathcal{L}\left(\boldsymbol{R}_1^{t-1}, \boldsymbol{\eta}_{\boldsymbol{x}}\right); \quad \boldsymbol{R}_1^{t} \leftarrow \arg\min_{\boldsymbol{R}_1} \mathcal{L}\left(\boldsymbol{R}_1, \boldsymbol{\eta}_{\boldsymbol{x}}^{t}\right) \tag{4}$$

where t represents the iteration index of the alternating rounds, and $\boldsymbol{\eta}_{\boldsymbol{x}}^{t}$ and $\boldsymbol{R}_1^{t}$ denote the values of $\boldsymbol{\eta}_{\boldsymbol{x}}$ and $\boldsymbol{R}_1$ at round t.

Solving for the cluster centroids $\boldsymbol{\eta}_{\boldsymbol{x}}$. The set of quantization parameters $\boldsymbol{g}_{\boldsymbol{x}}$ further contains the quantization scale $s_{\boldsymbol{x}}$ and zero point $z_{\boldsymbol{x}}$. In this paper, we adopt dynamic asymmetric per-token quantization for activations. Therefore, we can independently determine the optimal quantization scheme for solving $s_{\boldsymbol{x}}$ and $z_{\boldsymbol{x}}$ for each $\boldsymbol{x}\boldsymbol{R}_1$:

$$\begin{aligned} \boldsymbol{\eta}_{\boldsymbol{x}} = \mathcal{Q}_g(\boldsymbol{x}\boldsymbol{R}_1, s_{\boldsymbol{x}}^{t}, z_{\boldsymbol{x}}^{t}) = \text{clamp}\left(\left\lfloor \frac{\boldsymbol{x}\boldsymbol{R}_1}{s_{\boldsymbol{x}}} \right\rceil + z_{\boldsymbol{x}}, 0, 2^N - 1\right), \\ \text{where } s_{\boldsymbol{x}} = \frac{\max(\boldsymbol{x}\boldsymbol{R}_1) - \min(\boldsymbol{x}\boldsymbol{R}_1)}{2^N - 1}, z_{\boldsymbol{x}} = -\left\lfloor \frac{\min(\boldsymbol{x}\boldsymbol{R}_1)}{s_{\boldsymbol{x}}} \right\rceil \end{aligned} \tag{5}$$

where $\lfloor \cdot \rceil$ indicates round operation, N is the bitwidth.

Solving for $\boldsymbol{R}_1$. The right side od Eq 4 is well-known as Procrustes problem (Mulaik, 2009). which involves finding the optimal rotation matrix $\boldsymbol{R}_1$ that best aligns two sets of points, minimizing the Frobenius norm of their difference. The solution to this problem can be obtained through Singular Value Decomposition (SVD). Specifically, given input matrices $\boldsymbol{X}$ and its quantized version $\mathcal{Q}(\boldsymbol{X}, \boldsymbol{g})$, the optimal $\boldsymbol{R}_1$ can be found:

$$\boldsymbol{R}_1 = \boldsymbol{U}\boldsymbol{V}^T, \text{where } \boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}^T = \text{SVD}(\boldsymbol{X}^T \mathcal{Q}(\boldsymbol{X}, \boldsymbol{g}_{\boldsymbol{x}})). \tag{6}$$

where we treat the quantization parameters $\boldsymbol{g}^t$ as a constant.

One-step optimization. To find an improved rotation matrix $\boldsymbol{R}_1$ and quantization parameters $\boldsymbol{g}_{\boldsymbol{x}}$, we perform the iterative process shown in Eq 4. Specifically, a calibration set $\boldsymbol{X}^{cal}$ is randomly sampled from $\boldsymbol{X}$, the iterative process can be specified as:

$$s_{\boldsymbol{x}}^{t}, z_{\boldsymbol{x}}^{t} \leftarrow \arg\min_{s_x, z_x} \sum_{\boldsymbol{x} \in \boldsymbol{X}^{cal}} \left[\left\| \boldsymbol{x}\boldsymbol{R}_1^{t-1} - \mathcal{Q}_{s,z}(\boldsymbol{x}\boldsymbol{R}_1^{t-1}) \right\|_2^2 \right], \boldsymbol{\eta}_{\boldsymbol{x}}^{t} \leftarrow \mathcal{Q}_{s^t, z^t}(\boldsymbol{x}\boldsymbol{R}_1^{t-1}), \tag{7}$$

then the resulting quantization parameters will be used to produce the rotation matrix:

$$\boldsymbol{R}_1^{t} \leftarrow \arg\min_{\boldsymbol{R}_1} \sum_{\boldsymbol{x} \in \boldsymbol{X}^{cal}} \left[\left\| \boldsymbol{x}\boldsymbol{R}_1 - \boldsymbol{\eta}_{\boldsymbol{x}}^{t} \right\|_2^2 \right] \tag{8}$$

The detailed algorithm is provided in Algorithm 1.

4 Experiments

Experiment settings. We implemented DFRot based on QuaRot. In this paper, to simplify the problem, we apply dynamic asymmetric per-token quantization for activation values. The KV-cache is quantized using asymmetric quantization with a group size of 128. GPTQ (Frantar et al., 2022) are used for weight with per-channel symmetric quantization, where a linear search for the clipping ratio is applied to minimize squared error. We use a sample with sequence length of 2048 from WikiText-2 (Merity et al., 2016) training set to genrate calibration dataset $\boldsymbol{X}^{cal}$, initialize the rotation matrix $\boldsymbol{R}_1$ with RH, and optimize it for 100 iterations. After obtaining the optimized rotation matrix $\boldsymbol{R}_1$, we apply it to the corresponding model and achieve rotational invariance. We use 128 samples each with a sequence length of 2048, as the calibration dataset for GPTQ quantization.

#Bits W-A-KV	Method	LLaMA3-8B 0-shot[9] Avg.(↑)	LLaMA3-8B Wiki (↓)	LLaMA3-70B 0-shot[9] Avg.(↑)	LLaMA3-70B Wiki (↓)	LLaMA2-7B 0-shot[9] Avg.(↑)	LLaMA2-7B Wiki (↓)	LLaMA2-13B 0-shot[9] Avg.(↑)	LLaMA2-13B Wiki (↓)	LLaMA2-70B 0-shot[9] Avg.(↑)	LLaMA2-70B Wiki (↓)	LLaMA-7B 0-shot[9] Avg.(↑)	LLaMA-7B Wiki (↓)	LLaMA-13B 0-shot[9] Avg.(↑)	LLaMA-13B Wiki (↓)	LLaMA-30B 0-shot[9] Avg.(↑)	LLaMA-30B Wiki (↓)
16-16-16	FloatingPoint	68.09	6.14	73.81	2.86	65.21	5.47	67.61	4.88	71.59	3.32	64.48	5.68	66.67	5.09	70.00	4.10
4-4-16	RTN	33.42	6e2	31.21	8e3	32.44	nan	30.86	8e3	30.90	7e4	32.51	7e3	31.63	3e4	31.57	2e3
	SmoothQuant	33.04	1e3	34.67	2e2	32.13	nan	34.26	1e3	35.86	3e2	34.42	3e2	33.29	6e2	34.64	1e3
	GPTQ	32.98	5e2	31.47	4e4	32.72	nan	30.11	4e3	30.86	nan	32.12	1e3	31.51	3e3	30.88	2e3
	QuaRot	61.86	8.11	68.25	5.92	61.63	6.17	64.66	5.45	69.96	3.89	61.65	6.33	64.83	5.57	67.79	4.74
	DFRot	63.01	7.78	69.82	4.97	62.42	6.13	65.34	5.39	69.16	3.99	62.25	6.30	64.47	5.58	68.06	4.78
	SpinQuant*	64.11	7.28	66.99	6.10	57.37	6.78	63.23	5.24	70.58	3.68	61.82	6.08	64.59	5.36	68.08	4.53
	OSTQuant*	65.14	7.24	72.21	3.97	63.90	5.60	66.24	5.14	70.92	3.57	62.72	6.04	65.80	5.40	68.52	4.43
4-4-4	RTN	33.18	7e2	30.82	8e3	32.67	nan	30.93	7e3	31.73	7e4	32.87	1e4	31.33	3e4	31.64	2e3
	SmoothQuant	32.96	1e3	33.76	3e2	32.12	nan	33.36	1e3	35.54	3e2	33.32	3e2	33.28	5e2	34.65	1e3
	GPTQ	33.71	6e2	31.20	4e4	33.52	nan	27.85	5e3	31.09	nan	31.80	2e3	30.63	3e3	31.07	2e3
	Omniquant	32.33	4e2	-	-	48.40	14.26	50.35	12.30	-	-	48.46	11.26	45.63	10.87	45.04	12.35
	QuaRot	61.38	8.28	68.29	6.02	60.81	6.25	64.44	5.49	69.96	3.92	61.21	6.37	64.68	5.59	67.92	4.77
	DFRot	62.94	7.91	69.62	5.03	61.80	6.25	64.95	5.43	68.78	4.02	61.84	6.36	64.26	5.62	67.93	4.81
	SpinQuant*	64.10	7.35	66.31	6.24	62.01	5.96	64.13	5.74	70.57	3.61	61.32	6.12	64.95	5.39	68.14	4.55
	OSTQuant*	65.37	7.29	71.69	4.01	63.18	5.91	65.41	5.25	70.84	3.59	62.55	6.07	65.43	5.40	68.20	4.42

Table 2: Comparison of averaged accuracy on nine Zero-Shot tasks and perplexity on WikiText2. Results for SmoothQuant, GPTQ, OmniQuant, AWQ, SpinQuant and OSTQuant are from the OSTQuant paper, and QuaRot's results from the official code. * denotes the methods that use the quantization-aware training to optimize $\boldsymbol{R}_1$.

4.1 Main results

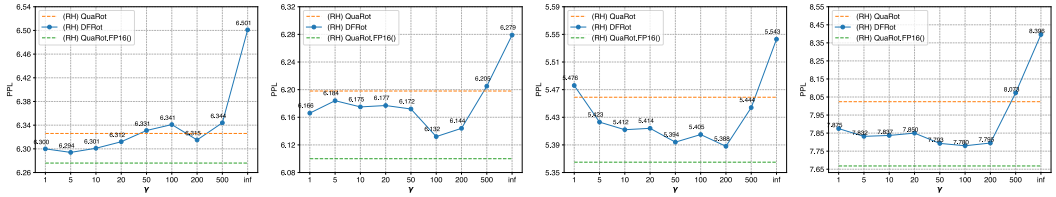
Language Generation Task. We evaluate DFRot on a language generation task and compare it with SmoothQuant (Xiao et al., 2023), GPTQ (Frantar et al., 2022), OmniQuant (Shao et al., 2023), AWQ (Lin et al., 2024a), SpinQuant (Liu et al., 2024) and OSTQuant (Hu et al., 2025). Table 2 shows the perplexity of LLaMA models. As shown, compared to QuaRot, DFRot achieves improvements in most cases. For example, DFRot achieves the most significant improvement on the LLaMA3-8B model with W4A4KV4 and W4A4KV16, outperforming QuaRot by 0.25 and 0.21, respectively. It is worth noting that DFRot has achieved near 1.00 PPL improvement on LLaMA3-70B, a model known for its challenging quantization performance, even surpassing SpinQuant, which finetunes $\boldsymbol{R}_1$ on wikitext through quantization-aware-training.

Similar to QuaRot, DFRot does not require any retraining process and only needs a sample to optimize the rotation matrix. On a single NVIDIA A100 80G GPU, it only takes an extra 8 minutes for LLaMA-7B & LLaMA2-7B & LLaMA3-8B and 20 minutes for LLaMA2-13B, resulting in minimal overhead. Even for the 70B models, the additional time is less than 90 minutes, which is also acceptable. It demonstrates that DFRot has wide applicability and can serve as a cost-effective post-training method to enhance the quantization performance of rotated LLMs. Although DFRot does not achieve the best performance compared to the state-of-the-art methods, like OSTQuant, we believe DFRot also help community to understand the fundamental performance gap between RO and RH.

Zero-Shot Tasks. We also evaluate DFRot on the following nine important zero-shot tasks: BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2021), OpenBookQA (Mihaylov et al., 2018), SIQA (Sap et al., 2019), HellaSwag (Zellers et al., 2019), Arc (Easy and Challenge) (Clark et al., 2018) and LAMBADA (Radford et al., 2019). We use lm_eval==0.4.5 (Gao et al., 2024) or our experiments. Table 2 shows the average score of DFRot on the above tasks. As can be seen, DFRot consistently achieves improvements compared to QuaRot across all tasks. For example, DFRot achieves a 1.56% accuracy improvement compared to QuaRot on the LLaMA3-8B model with W4A4KV4 quantization settings.

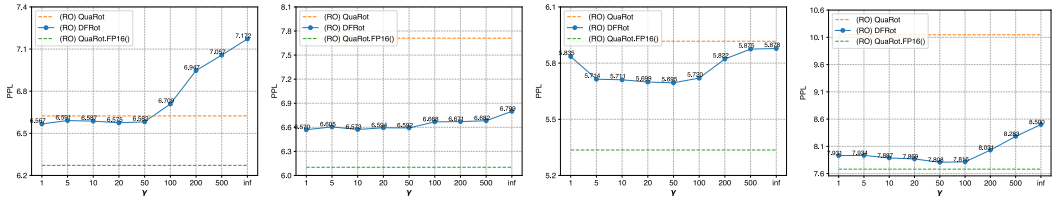
4.2 Ablation studies

Choice of γ. To further understand the effect of hyperparameters in DFRot, we conducted an ablation study on Wikitext-2 PPL to investigate the impact of different γ settings for W4A4KV16. As seen in Figure 5, when γ ranges between 50 and 200, DFRot achieves significant improvements across various LLaMA models using RH. Notably, on the LLaMA3-8B



(a) LLaMA-7B (b) LLaMA2-7B (c) LLaMA2-13B (d) LLaMA3-8B

Figure 5: Comparison of WikiText-2 perplexity results under different γ for W4A4KV16. $\boldsymbol{R}_1$ is initialized with RH.



(a) LLaMA-7B (b) LLaMA2-7B (c) LLaMA2-13B (d) LLaMA3-8B

Figure 6: Comparison of WikiText-2 perplexity results under different γ for W4A4KV16. $\boldsymbol{R}_1$ is initialized with RO.

model, which is known for its quantization performance sensitiveness to massive activations from Table 1, we observed a PPL improvement of over 0.2 in Figure 5(d). If we set $\gamma = 1$ and treat $\boldsymbol{X}^m$ and $\boldsymbol{X}^{cal} \setminus \boldsymbol{X}^m$ equally to minimize their quantization errors, it may reduce the quantization loss of $\boldsymbol{X}^{cal} \setminus \boldsymbol{X}^m$ but increase the quantization loss of $\boldsymbol{X}^m$, ultimately resulting in a performance decline on the LLaMA2-13B. Conversely, if we set $\gamma \to \infty$ and only optimize the quantization error for $\boldsymbol{X}^m$, it will increase the quantization error of $\boldsymbol{X}^{cal} \setminus \boldsymbol{X}^m$, resulting in an accuracy drop across the LLaMA-7B, LLaMA2-7B, LLaMA2-13B and LLaMA3-8B models.

Initialize with Randomized Orthogonal. We conducted an ablation to study the effectiveness of DFRot when $\boldsymbol{R}_1$ initialized with RO. We keep the same experimental settings as in the study with RH and optimize the rotation matrix with different γ values. As shown in Figure 6, our method achieves considerable improvements in RO scenarios compared to using RH for initialization. Meanwhile, it is more effective for LLM whose quantization performance is more sensitive to the massive activations, such as LLaMA3-8B and LLaMA3-70B. However, due to the exceptional performance of RH, initialization and optimization using RH always yield superior final results compared to those obtained with RO. Therefore, we recommend using RH for initialization in practice to achieve better performance.

Model	**Sample1 (64×2048)**	**Sample2 (64×2048)**	**Sample3 (64×2048)**	**Sample4 (64×2048)**	**Sample5 (64×2048)**
LLaMA3-8B	7.78	7.76	7.79	7.74	7.76
LLaMA2-7B	6.13	6.12	6.15	6.11	6.14
Model	**Sample1 (48×2048)**	**Sample1 (32×2048)**	**Sample1 (24×2048)**	**Sample1 (16×2048)**	**Sample1 (8×2048)**
LLaMA3-8B	7.78	7.78	7.77	7.80	7.86
LLaMA2-7B	6.14	6.15	6.15	6.12	6.20

Table 3: Comparison of WikiText-2 perplexity results under different calibration samples for W4A4KV16.

4.3 Analysis of Calibration Set Sensitivity

We performed ablation studies W4A4KV16 on LLaMA3-8B and LLaMA2-7B along two dimensions: the choice of calibration samples and the num of calibration tokens and evaluate

results on WikiText. Samples are all sampled from WikiText-2 train. In selecting the number of tokens, we utilize the inputs to the first N transformer blocks as the calibration data source and demonstrate results in Table 3. For example, when 48×2048 tokens are selected, the inputs to transformer blocks 0 to 23 are used for calibration. Our results indicate that, for tuning the rotation matrices in LLaMA3-8B and LLaMA2-7B, using 16×2048 tokens is often sufficient. We believe that these reasons may all be the causes for the optimization of DFRot being relatively insensitive to the number of tokens in the calibration dataset:

1. These special tokens will appear in relatively shallow network layers (Sun et al., 2024), therefore, a small number of layers are also sufficient to capture these tokens.
2. For a model, tokens with massive activations in LLMs tend to exhibit only a few similar data distributions because these tokens are often produced by out_proj or down_proj layers with large weights (Yu et al., 2024).
3. GPTQ will use 128 samples with a length of 2048 to calibrate the weights, which reduces the impact of the sample size during rotation matrix calibration.

4.4 Results on MMLU

W-A-KV	Methods	LLaMA2-7B	LLaMA3-8B	QWen2-7B	Mistral-7B-v0.3
16-16-16	FP	41.85	62.23	69.47	59.11
4-4-16	QuaRot	34.83	51.43	62.67	52.82
4-4-16	DFRot	35.54	51.68	63.40	53.38

Table 4: Comparison of MMLU results under different methods.

We compare DFRot with QuaRot with W4A4KV16 quantization configuration with different models. As seen in Table 4, even though rotation matrix $\boldsymbol{R}_1$ is refined with WikiText-2 dataset, DFRot also outperforms QuaRot in all models. It indicates that DFRot, which refines $\boldsymbol{R}_1$ by optimized long tailed quantization error, can be seen as a general method. It is also worth noting that even though DFRot achieves slight improvement with WikiText2 for LLaMA2-7B, it achieves 0.71% improvement with MMLU, which is significant. On the contrary, for the LLaMA3-8B, while DFRot achieves significant improvement with WikiText2, it only achieves 0.25% improvement with MMLU, which is slight. To sum up, we can know that the PPL with WikiText2 can not been seen as a good indicator of the model downstream performance. In the future, we will study how to design more robust quantization algorithms for downstream tasks to further enhance the capabilities of quantized models in downstream tasks.

5 Conclusion

Eliminating outliers in LLMs through rotational invariance can significantly improve model quantization accuracy. In this paper, we find that in the context of 4-bit activation quantization, the fundamental reason for the effectiveness difference between RO and RH is their performance on tokens with massive activations. Specifically, randomized Hadamard transformations perform better on these tokens than random Orthogonal transformation. Based on the observation that tokens with massive activations are rare and important in LLMs, we treat the problem as a long-tail optimization and construct a simple yet effective weighted quantization loss function to balance the importance of tokens. Furthermore, by alternately employing orthogonal Procrustes transformations to refine the rotation matrix $\boldsymbol{R}_1$ and optimizing quantization parameters for $\boldsymbol{X}$, our method, named DFRot, enhances the Rotated LLMs by achieving Dual Free, including *Outlier-Free* and *Massive Activation-Free*. It is worth noting that DFRot significantly improves model accuracy in 4-bit activation quantization with just a single data sample, achieving PPL improvements of 0.98 and 0.95 on W4A4KV4 and W4A4KV16, respectively, for the LLaMA3-70B, which is notable for its quantization challenge.

References

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

Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. Slicegpt: Compress large language models by deleting rows and columns. *arXiv preprint arXiv:2401.15024*, 2024a.

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

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 7432–7439, 2020.

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

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

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.

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.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pp. 10323–10337. PMLR, 2023.

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

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024. URL https://zenodo.org/records/12608602.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Xing Hu, Yuan Cheng, Dawei Yang, Zukang Xu, Zhihang Yuan, Jiangyong Yu, Chen Xu, Zhe Jiang, and Sifan Zhou. Ostquant: Refining large language model quantization with orthogonal and scaling transformations for better distribution fitting. *arXiv preprint arXiv:2501.13987*, 2025.

Wei Huang, Xudong Ma, Haotong Qin, Xingyu Zheng, Chengtao Lv, Hong Chen, Jie Luo, Xiaojuan Qi, Xianglong Liu, and Michele Magno. How good are low-bit quantized llama3 models? an empirical study. *arXiv preprint arXiv:2404.14047*, 2024.

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

Fengyu Li, Yilin Li, Junhao Zhu, Lu Chen, Yanfei Zhang, Jia Zhou, Hui Zu, Jingwen Zhao, and Yunjun Gao. Aistorian lets ai be a historian: A kg-powered multi-agent system for accurate biography generation. *arXiv preprint arXiv:2503.11346*, 2025.

Jun Li, Li Fuxin, and Sinisa Todorovic. Efficient riemannian optimization on the stiefel manifold via the cayley transform. *arXiv preprint arXiv:2002.01113*, 2020.

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

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

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

J Macqueen. *Some methods for classification and analysis of multivariate observations*. University of California Press, 1967.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Yuhong Mo, Hao Qin, Yushan Dong, Ziyi Zhu, and Zhenglin Li. Large language model (llm) ai text generation detection based on transformer deep learning algorithm. *arXiv preprint arXiv:2405.06652*, 2024.

Stanley A Mulaik. *Foundations of factor analysis*. CRC press, 2009.

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.

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.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

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.

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

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

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

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.

Bohao Wang, Feng Liu, Changwang Zhang, Jiawei Chen, Yudi Wu, Sheng Zhou, Xingyu Lou, Jun Wang, Yan Feng, Chun Chen, et al. Llm4dsr: Leveraing large language model for denoising sequential recommendation. *arXiv preprint arXiv:2408.08208*, 2024.

Bohao Wang, Feng Liu, Jiawei Chen, Xingyu Lou, Changwang Zhang, Jun Wang, Yuegang Sun, Yan Feng, Chun Chen, and Can Wang. Msl: Not all tokens are what you need for tuning llm as a recommender. *arXiv preprint arXiv:2504.04178*, 2025.

Jiateng Wei, Quan Lu, Ning Jiang, Siqi Li, Jingyang Xiang, Jun Chen, and Yong Liu. Structured optimal brain pruning for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 13991–14007, 2024.

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 optimal shifting and scaling. *arXiv preprint arXiv:2304.09145*, 2023.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. A survey on large language models for recommendation. *CoRR*, abs/2305.19860, 2023.

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

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems*, 35:27168–27183, 2022.

Mengxia Yu, De Wang, Qi Shan, and Alvin Wan. The super weight in large language models. *arXiv preprint arXiv:2411.07191*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

Biao Zhang, Barry Haddow, and Alexandra Birch. Prompting large language model for machine translation: A case study. In *International Conference on Machine Learning*, pp. 41092–41110. PMLR, 2023.

Ying Zhang, Peng Zhang, Mincong Huang, Jingyang Xiang, Yujie Wang, Chao Wang, Yineng Zhang, Lei Yu, Chuan Liu, and Wei Lin. Qqq: Quality quattuor-bit quantization for large language models. *arXiv preprint arXiv:2406.09904*, 2024.

Maosen Zhao, Pengtao Chen, Chong Yu, Yan Wen, Xudong Tan, and Tao Chen. Pioneering 4-bit fp quantization for diffusion models: Mixup-sign quantization and timestep-aware fine-tuning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18134–18143, 2025.