MPPQ: Enhancing Post-Training Quantization for LLMs via Mixed Supervision, Proxy Rounding, and Pre-Searching

Mingrun Wei[1] , **Yeyu Yan**[1] , **Dong Wang**[1*]

[1]School of Computer Science and Technology, Beijing Jiaotong University, Beijing, China
weimingrun@bjtu.edu.cn, yanyeyu-work@foxmail.com, wangdong@bjtu.edu.cn

Abstract

Recently, post-training quantization (PTQ) methods for large language models (LLMs) primarily focus on tackling the challenges caused by outliers. Scaling transformation has proven to be effective while how to enhance the performance of extremely low-bitwidth (e.g., 2-bit) PTQ under it remains largely unexplored. In this work, a new PTQ framework, namely **MPPQ**, is established. Specifically, MPPQ first proposes an enhanced reconstruction loss based on ***Mixed metric supervision*** to mitigate the distribution inconsistency caused by quantization while providing strong regularization for learnable parameters. Secondly, we introduce a ***Proxy-based adaptive rounding*** scheme in weight quantization, which replaces the round-to-nearest (RTN) function to minimize the overall quantization errors through element-wise scaling. Furthermore, a ***factor coarse Pre-searching*** mechanism is presented to ensure proper coordination between quantization and clipping patterns, while achieving optimal initialization of clipping factors before training. Extensive experiments show that MPPQ consistently outperforms state-of-the-art methods in low-bit quantization settings. For instance, the perplexity of WikiText2 can be dramatically reduced to 8.85 (3.9 $\downarrow$ vs 12.75 of the latest method, LRQuant) for the LLaMA-2-7B model, which is quantized with W4A4.

1 Introduction

Large language models (LLMs) [OpenAI, 2023; Touvron *et al.*, 2023a; Touvron *et al.*, 2023b] excel at complex natural language processing (NLP) tasks, yet they also come with high computational resource and parameter storage demands. Consequently, it is almost not feasible to deploy LLMs on constrained hardware. Quantization, especially post-training quantization (PTQ) [Dettmers *et al.*, 2022; Xiao *et al.*, 2023; Shao *et al.*, 2024], has recently gained preference from both industry and academia due to its ability to operate with limited computational overhead and calibration data.

*Dong Wang is the corresponding author

The PTQ of LLMs presents unique challenges, along with the inherent flaws of the technique. On the one hand, outlier channels [Dettmers *et al.*, 2022] in the intermediate activations of LLMs are matters that need to be taken seriously. A feasible solution [Xiao *et al.*, 2023; Wei *et al.*, 2023; Shao *et al.*, 2024; Zeng *et al.*, 2024; Zhao *et al.*, 2024] is to migrate it to the corresponding weights, and then apply clipping in weight quantization to relieve the impact of outliers. In terms of quantization tuning, prevalent PTQ methods [Shao *et al.*, 2024] for LLMs generally utilize the mean squared error (MSE) [Choukroun *et al.*, 2019] between the outputs of full-precision and quantized blocks as a supervised signal for updating learnable parameters. This optimization technique, however, offers limited supervision and falls short in aligning block-level and layer-level output activations. On the other hand, AdaRound [Nagel *et al.*, 2020] demonstrated that the round-to-nearest (RTN) operation is suboptimal and proposed a learnable quantization rounding in the form of element-wise addition. Meanwhile, recent progress [Han *et al.*, 2015; Lee *et al.*, 2023b; Lee *et al.*, 2024] also argued that weights with large magnitudes are relatively important and should be allowed to be quantized to discrete values further from themselves. Clearly, existing PTQ for LLMs methods [Xiao *et al.*, 2023; Shao *et al.*, 2024; Ma *et al.*, 2024; Zeng *et al.*, 2024; Zhao *et al.*, 2024; Liu *et al.*, 2024a] have not yet given sufficient consideration to this. Moreover, [Gong *et al.*, 2024] suggested that asymmetric quantization should be paired with asymmetric clipping, and vice versa. Unfortunately, previous methods often incorrectly initialized the weight clipping factor, leading to an abnormal combination of quantization and clipping patterns.

To this end, we propose MPPQ, an accurate and efficient low-bitwidth PTQ framework tailored for LLMs. Specifically, MPPQ first introduces a novel Mixed Metric Supervision (MMS) approach that integrates multi-granularity and multi-dimensional regularization into block-wise reconstruction. In terms of granularity, it includes both block-wise and layer-wise loss, and in terms of dimension, it considers both magnitude similarity and direction similarity. Such approach enhances the consistency of quantized blocks with full-precision while also leading to more robust parameter learning. Inspired by the Low-Rank Adaptation (LoRA) [Hu *et al.*, 2022] paradigm, we propose a Proxy-based Adaptive Rounding (PAR) scheme that employs two learnable low-

rank matrices and optimizes them under the MMS. By performing the Hadamard product of the low-rank matrices with the corresponding weights, each element of the weights can be modified with varying intensities relative to their individual magnitudes. Notably, this scheme rectifies the inaccuracies caused by the RTN operator and leverages the efficiency of LoRA fine-tuning. Furthermore, our MPPQ proposes a factor coarse pre-searching (FCP) mechanism. The mechanism engages a grid search algorithm to iteratively identify the optimal clipping factors based on minimizing quantization errors across each weight layer, before the training begins. In this way, the proper combination of quantization and clipping patterns is ensured, while also promoting the stable convergence of weight clipping factors in optimization.

The contributions of this paper are summarized as follows:

1. We propose a novel mixed metric supervision approach under block-wise reconstruction, ensuring the distributions of quantized blocks align with full-precision counterparts while providing better regularization for parameter learning.
2. A proxy-based adaptive rounding scheme for weight quantization is constructed with the aim of reducing the quantization error caused by the RTN operator without incurring any extra inference overhead.
3. We introduce a factor coarse pre-searching mechanism for better weight clipping, which ensures the correct combination of quantization and clipping patterns and provides a better initial learning state for parameters.
4. Extensive experiments across models and datasets verify the robust performance of MPPQ, particularly in extremely low quantization bitwidths (e.g., W4A4, W6A6, and W2A16) scenarios. For example, our average accuracy for the 4/4-bit LLaMA-2-7B model is 54.87% (2.17% ↑ vs 52.70% in ABQ-LLM). Similarly, on the W2A16 configuration of LLaMA-13B, our perplexity on the C4 dataset is 9.79 (2.67 ↓ vs 12.46 in AffineQuant).

2 Related Work

2.1 Network Quantization

Quantization means to represent the weights, activations, and even gradients with low-bitwidth, leading to smaller models and faster inference. This technique can be largely divided into two categories: quantization-aware training (QAT) and post-training quantization. Thanks to the straight-through estimator (STE) [Bengio *et al.*, 2013], QAT is capable of meeting most of the demands but requires high GPU effort and leverages the entire dataset. PTQ offers a more friendly alternative, enabling to quickly determine the optimal quantizer for a network using a small amount of data and has recently brought competitive outcomes. AdaRound [Nagel *et al.*, 2020] showed that the RTN operation is not often the optimal solution and proposed reconstructing in a manner based on per-layer learning. BRECQ [Li *et al.*, 2021] extended AdaRound to per-block reconstruction by utilizing the Fisher information. [Wei *et al.*, 2022a] proposed QDrop, which randomly discards quantized activation values during PTQ to further enhance low-bit performance. PD-Quant [Liu *et al.*, 2023] incorporated global prediction differences into reconstruction and adjusted the distribution of activations to alleviate the issue of overfitting.

2.2 Quantization on LLMs

LLMs quantization can be broadly divided into weight-only quantization [Frantar *et al.*, 2022; Lin *et al.*, 2024; Cheng *et al.*, 2023; Kim *et al.*, 2024; Chee *et al.*, 2024; Lee *et al.*, 2023a] and weight-activation quantization [Dettmers *et al.*, 2022; Yao *et al.*, 2022; Yuan *et al.*, 2023; Xiao *et al.*, 2023; Wei *et al.*, 2023; Shao *et al.*, 2024; Liu *et al.*, 2024b; Zeng *et al.*, 2024; Zhao *et al.*, 2024] based on the quantization objects. The former typically employs techniques such as compensation, scaling, rotation, and mixed-precision to compress weights. However, matrix operations at the hardware level are still executed with high precision, meaning that weight-only quantization fails to truly accelerate the inference of LLMs. Conversely, the latter can invoke specific hardware units during the inference phase, but outliers in activations are also a tough issue that must be addressed. To this end, SmoothQuant [Xiao *et al.*, 2023] utilized mathematical equivalent transformations to shift outliers to corresponding weights. [Wei *et al.*, 2023] further expanded channel-wise shifting while [Shao *et al.*, 2024] employed a differentiable approach to learn the optimal parameters mentioned above, as well as weight clipping factors. QLLM [Liu *et al.*, 2024a] proposed to reallocate the outliers to other channels to mitigate their impact on the quantization range. LRQuant [Zhao *et al.*, 2024] sought better quantization performance by refining the initialization of the learnable smoothing parameters. In a notable difference, our proposed MPPQ focuses on loss construction, rounding optimization, and pattern matching to overcome the poor performance of existing works in extremely low-bitwidth quantization.

3 Preliminaries

3.1 *b*-bit Quantization

In this paper, we adopt hardware-friendly uniform asymmetric quantization for both weights and activations. For the matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$ with floating-point values and a quantization bitwidth of b, the quantization and de-quantization procedures can be expressed as follows:

$$\hat{\mathbf{W}}_{ij} = s \left[\text{clamp}(\left\lfloor \frac{\mathbf{W}_{ij}}{s} \right\rceil + zp; 0, 2^b - 1) - zp \right], \quad (1)$$

where s and zp represent the quantization step size and the zero point, respectively:

$$s = \frac{\max(\mathbf{W}) - \min(\mathbf{W})}{2^b - 1}, zp = - \left\lfloor \frac{\min(\mathbf{W})}{s} \right\rceil. \quad (2)$$

$\lfloor \cdot \rceil$ represents the RTN function and $\text{clamp}(x, x_{\text{min}}, x_{\text{max}})$ denotes the operation that constrains the value x within $[x_{\text{min}}, x_{\text{max}}]$. Weight-only quantization enables memory footprint savings, while weight-activation quantization can further accelerate inference. Previous studies [Dettmers *et al.*, 2022; Xiao *et al.*, 2023; Wei *et al.*, 2022b] have revealed that several outlier channels are prevalent in the activations of

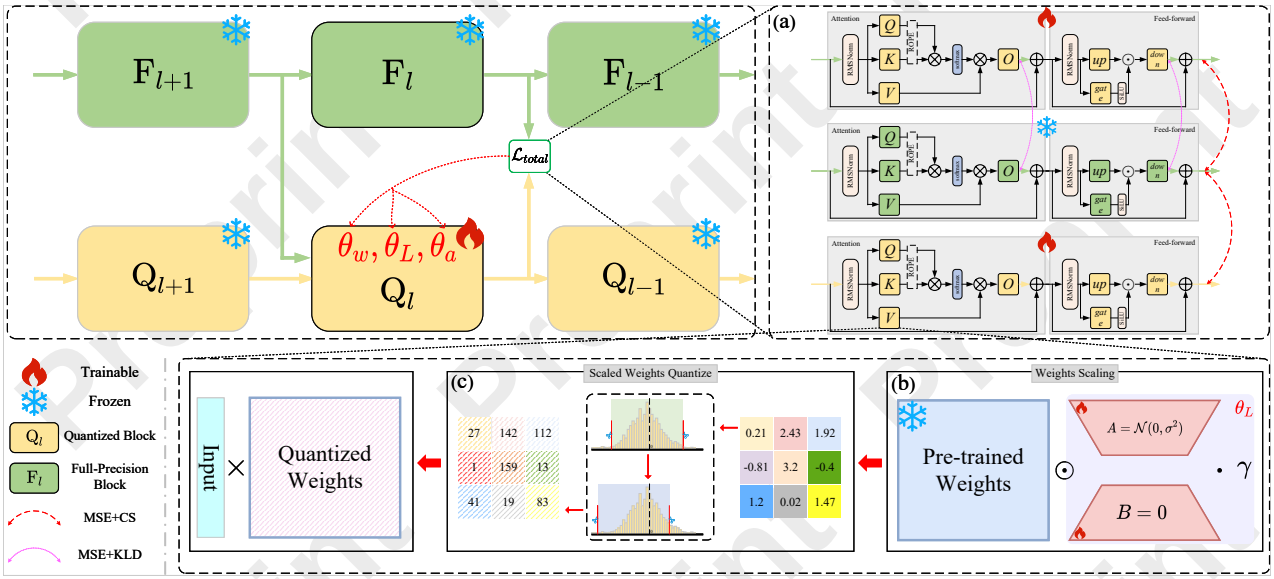


Figure 1: The overall framework of MPPQ. (a): Mixed metric supervision approach for block-wise reconstruction. The green and yellow arrowed lines represent data flows, while the green and yellow rectangles denote full-precision and quantized transformer blocks, respectively. (b): The proxy-based adaptive rounding scheme leveraging the LoRA structure. (c): An exemplary view of the proposed factor coarse pre-searching mechanism.

LLMs, which enlarges the quantization range and thus hinders the accurate mapping of floating-point values with varying magnitudes. Therefore, the current consensus is that what must be done before activation quantization is to suppress the widespread outliers.

3.2 Scaling Transformation

Given the input activation $\mathbf{X}^{l-1} \in \mathbb{R}^{m \times n}$, a mainstream approach to suppress outliers is to first determine a diagonal matrix $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$, based on the heuristic principle. Then, left-multiply it by the weight and right-multiply the activation with its inverse, respectively, to ensure computational invariance.

$$\mathbf{X}^l = (\mathbf{X}^{l-1}\mathbf{\Lambda}^{-1})(\mathbf{\Lambda}\mathbf{W}^\top). \tag{3}$$

By inserting such transformations into LLMs, the impact of outliers on the quantization errors is effectively mitigated. In parallel, $\mathbf{\Lambda}$ can be incorporated into $\mathbf{W}^\top$, thereby eliminating the runtime burden. The diagonal element $\mathbf{\Lambda}_i$ within $\mathbf{\Lambda}$ is computed by the following formula:

$$\mathbf{\Lambda}_i = \max(|\mathbf{X}_i|)^\alpha / \max(|\mathbf{W}_i|)^{1-\alpha}, \tag{4}$$

where α represents the migration strength and is often set manually. Building on this point, [Wei *et al.*, 2023] further introduced an additional channel-wise shifting operation:

$$\mathbf{X}^l = (\mathbf{X}^{l-1} - \delta) \oslash \Delta \cdot \Delta \odot \mathbf{W}^\top + \delta \mathbf{W}^\top, \tag{5}$$

where $\delta \in \mathbb{R}^{1 \times n}$ and $\Delta \in \mathbb{R}^{1 \times n}$ represent matrices composed of per-channel shifting and scaling, respectively.

4 Approach

In this section, we present the details of our proposed quantization framework. The overall pipeline of MPPQ is illustrated in Figure 1 and mainly consists of three components. MPPQ adheres to scaling transformation [Xiao *et al.*, 2023; Wei *et al.*, 2023; Shao *et al.*, 2024] for handling outliers in activation, then it performs the Hadamard product with the weight and two learnable low-rank matrices, in anticipation of altering the quantization rounding of the weights. This is followed by determining the optimal initial values of weight clipping factors using grid search. Finally, MPPQ utilizes the proposed mixed metric supervision to calculate quantization loss and update all learnable parameters until convergence.

4.1 Mixed Metric Supervision

Previous findings [Li *et al.*, 2021; Wei *et al.*, 2022a] have suggested that it is necessary to optimize layers with dependencies together in PTQ. Based on this, a block-wise quantization reconstruction paradigm has been widely applied in current PTQ methods for LLMs. The paradigm typically employs MSE to compute the loss between the outputs of the full-precision and quantized blocks, and then uses it as the sole metric to guide the quantization process:

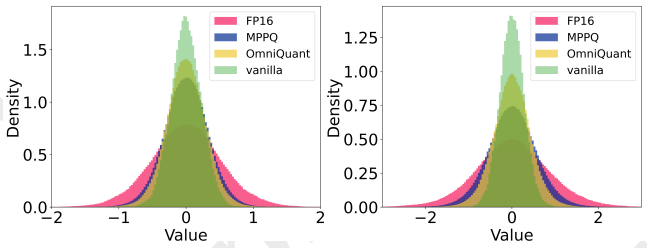
$$\min_{\theta_w, \theta_a} \left\| \mathcal{F}(\mathbf{W}, \mathbf{X}_{fp}) - \mathcal{F}(Q_w(\mathbf{W}; \theta_w), Q_a(\mathbf{X}_q; \theta_a)) \right\|_F^2, \tag{6}$$

where $\mathcal{F}(\cdot)$ represents the mapping function for a transformer block in LLMs, $Q_w(\cdot)$ and $Q_a(\cdot)$ are the weight and activation quantizers, respectively, with parameters θ_w and θ_a. $\|\cdot\|_F$ denotes the Frobenius Norm.

Eq. (6) can be further simplified to:

$$\min_{\theta} \mathcal{L}_{MSE}(\mathcal{F}_{fp}(\mathbf{X}_{fp}), \mathcal{F}_q(\mathbf{X}_q; \theta)), \tag{7}$$

where $\mathbf{X}_{fp}$ and $\mathbf{X}_q$ are the inputs of the full-precision and the quantized block, respectively, which also correspond to the outputs of their respective previous blocks. Weights are omitted in Eq. (7).



(a) $18th$ transformer block. (b) $23th$ transformer block.

Figure 2: Visualization of the output distribution of different blocks from LLaMA-2-7B on WikiText2 under W4A4 quantization.

As observed in Figure 2, the significant difference between the output distributions of full-precision and quantized blocks suggests that relying solely on MSE loss for block-wise quantization reconstruction may not be sufficient to effectively align them. To address this, we propose a novel reconstruction loss function based on mixed metric supervision to impose stronger constraints on the distributional similarity between the layer-wise and block-wise outputs, thereby mitigating quantization errors.

Concretely, we first align the input of the quantized transformer block with its full-precision counterpart and devise a novel variant that differs from the losses used in prior works:

$$\min_{\theta} \mathcal{L}_{MSE}(\mathcal{F}_{fp}(\mathbf{X}_{fp}), \mathcal{F}_{q}(\mathbf{X}_{fp}; \theta)). \tag{8}$$

Next, for Eq. (7) and Eq. (8), we add cosine similarity (CS) loss to capture the directional differences as well as to achieve a more robust parameter optimization.

Furthermore, we impose additional constraints on the feature representation of certain layers within the quantized block, ensuring that its overall distribution is closer to that of the full-precision block while helps to smooth outliers in the feature space. In the end, we integrate these loss functions to formulate the final optimization objective $\mathcal{L}_{total}$ as follows:

$$\begin{aligned}\min_{\theta} \mathbb{E}\ \Big[&\mathcal{O}^{(1)}(\mathcal{F}(\mathbf{W}, \mathbf{X}_{fp}), \mathcal{F}(Q_w(\mathbf{W}; \theta_w, \theta_L), Q_a(\mathbf{X}_q; \theta_a)) \\ &+ \mathcal{O}^{(2)}(\mathcal{F}(\mathbf{W}, \mathbf{X}_{fp}), \mathcal{F}(Q_w(\mathbf{W}; \theta_w, \theta_L), Q_a(\mathbf{X}_{fp}; \theta_a))\Big],\end{aligned} \tag{9}$$

where $\mathcal{O}^{(1)}$ and $\mathcal{O}^{(2)}$ represent the loss functions. The common procedure of both, as well as the part unique to $\mathcal{O}^{(2)}$ are as follows, respectively:

$$\mathcal{K}(\mathbf{F}, \mathbf{Q}) = \|\mathbf{F} - \mathbf{Q}\|_F^2 + \frac{1}{n}\sum_{j=1}^{n}(1 - \frac{\mathbf{F}^j \cdot \mathbf{Q}^j}{\|\mathbf{F}^j\|_2 \|\mathbf{Q}^j\|_2}), \tag{10}$$

$$\begin{aligned}\mathcal{H}(\mathbf{f}, \mathbf{q}) = \|\mathbf{f}_i - \mathbf{q}_i\|_F^2 + \mathrm{D}_{KL}(\sigma(\mathbf{f}_i), \sigma(\mathbf{q}_i)), \\ \mathbf{s.t.}\ i \in \{\mathrm{o\_proj}, \mathrm{down\_proj}\}.\end{aligned} \tag{11}$$

In which, $\mathbf{F}$ and $\mathbf{Q}$ denote the full-precision and quantized block outputs, with $\mathbf{f}_i$ and $\mathbf{q}_i$ representing the full-precision and quantized outputs at layer i, respectively. $\|\cdot\|_2$ denotes the Euclidean Norm. $\mathrm{D}_{KL}(\cdot)$ represents the Kullback-Leibler divergence [Kullback and Leibler, 1951] and σ is the softmax function.

4.2 Proxy-Based Adaptive Rounding

PTQ typically assigns each weight to its nearest fixed-point representation, whereas AdaRound [Nagel *et al.*, 2020] has demonstrated that this approach is not optimal. It introduced a method to adaptively determine the best rounding for each weight of the model through back-propagation, as illustrated below:

$$\hat{\mathbf{W}}_{ij} = s\left[\mathrm{clamp}\left(\left\lfloor\frac{\mathbf{W}_{ij}}{s}\right\rfloor + h(\mathbf{V}_{ij}) + zp, n, p\right) - zp\right], \tag{12}$$

where $\mathbf{W}$ and $\hat{\mathbf{W}}$ represent the pre-trained weight and its quantized version, respectively. $h(\mathbf{V}) \in \{0,1\}^{m\times n}$ is the learnable rounding matrix and its size must be consistent with $\mathbf{W}$. Directly applying AdaRound to LLMs quantization is not feasible, as LLMs typically have billions of parameters, which makes solving $h(\mathbf{V})$ on a small calibration dataset challenging. Moreover, previous studies [Lee *et al.*, 2024; Han *et al.*, 2015] have demonstrated that the fixed-point representations of weights with large magnitudes in LLMs should be allowed to deviate more from their floating-point values. Therefore, there is an urgent need to propose a new rounding method within the aforementioned limitations.

Inspired by the parameter sharing of the LoRA structure, we explore a proxy-based adaptive rounding scheme for weight quantization. As shown in Figure 1, two learnable low-rank matrices are deployed and their product serves as the proxy rounding matrix, which is then merged with the weight by performing the Hadamard product. PAR achieves element-wise scaling of weights with varying intensities prior to quantization, and thus is able to adjust the quantization rounding values accordingly for each weight.

The aforementioned process can be summarized as follows:

$$\hat{\mathbf{W}}_{ij} = s\left[\mathrm{clamp}\left(\left\lfloor\frac{\mathbf{W}_{ij} \odot \mathbf{P}_{ij} \cdot \gamma}{s}\right\rfloor + zp, n, p\right) - zp\right]. \tag{13}$$

In which, $\mathbf{P}$ is the proxy rounding matrix that is defined as $\mathbf{P}_{ij} = \exp(\mathbf{B}\cdot\mathbf{A})_{ij}$. $\mathbf{B} \in \mathbb{R}^{m\times r}$ and $\mathbf{A} \in \mathbb{R}^{r\times n}$ are low-rank matrices with a rank of r and initialized by random Gaussian and zeros, respectively. γ serves for balancing.

On the flip side, the gradient of $\mathcal{L}_{total}$ with respect to $\mathbf{P}_{ij}$ can be derived as follows[1]:

$$\frac{\partial \mathcal{L}_{total}}{\partial \mathbf{P}_{ij}} = \frac{\partial \mathcal{L}_{total}}{\partial \hat{\mathbf{W}}} \frac{\partial \hat{\mathbf{W}}}{\partial \mathbf{P}_{ij}} \simeq (\mathbf{W}_{ij} \cdot \gamma)\frac{\partial \mathcal{L}_{total}}{\partial \hat{\mathbf{W}}}. \tag{14}$$

As expected, Eq. (14) indicates that the scaling degree of each pre-trained weight is proportional to its own magnitude, which matches the findings of [Lee *et al.*, 2023b]. In other words, the larger $\mathbf{W}_{ij}$ is, the greater the chance that $\hat{\mathbf{W}}_{ij}$ deviates from $\mathbf{W}_{ij}$. Notably, after the reconstruction, the low-rank matrices can be absorbed by the network without introducing any extra inference costs.

[1]The detailed proof can be found in Appendix A.

Algorithm 1 Factor Coarse Pre-searching

Input: Pre-trained FP transformer block
Parameter: Number of weight layers N, grid search iterations m, starting search point p, initial loss and clipping factors $\mathcal{L}^*$, θ^*_{max}, and θ^*_{min}
Output: Optimal clipping factors θ^*_{max} and θ^*_{min}

```
Let L* to +∞, θ*_max and θ*_min to −1.
while n < N do
    while j < m do
        Calculate the θ_max and v_max by Eq. (15) and Eq. (16).
        while k < m do
            Calculate the θ_min and v_min by Eq. (15) and Eq. (17).
            Quantize weight layer n based on Eq. (1).
            Calculate L using Eq. (18).
            if L* > L then
                L* ← L, θ*_max ← θ_max, θ*_min ← θ_min.
            end if
        end while
    end while
end while
return θ*_max and θ*_min
```

4.3 Factor Coarse Pre-searching

Reducing errors caused by outliers in weights or activations in the quantization of LLMs is crucial. Previous studies, including [Shao *et al.*, 2024], have implemented weight clipping with clipping factors that are symmetrically initialized by hand. Obviously, this method fails to consider the crucial compatibility between quantization and clipping patterns, as well as whether the clipping factors are in the optimal initial state before training, which is also highly questionable. In this section, we present a factor coarse pre-searching mechanism for weight quantization, which can serve as a pre-operation and skillfully overcome the aforementioned issues.

The running procedure of the mechanism is summarized in Algorithm 1 and can be roughly divided into four steps. We first exploit a grid search algorithm and set its hyper-parameters, including the total number of iteration rounds m and the starting search point denoted as $p \in [0, 1]$. Next, the maximum and minimum values of all quantization intervals for the weight $\mathbf{W}$ to be updated under the current iteration round $\lambda \in [0, m)$.

$$\theta^i_{max}, \theta^i_{min} = p + \frac{1-p}{m}\lambda, \tag{15}$$

$$v^i_{max} = \theta^i_{max} \times \max(\mathbf{W}_i), \tag{16}$$

$$v^i_{min} = \theta^i_{min} \times \min(\mathbf{W}_i), \tag{17}$$

where $\mathbf{W}_i$ represents the i-th quantization interval. θ^i_{max} and θ^i_{min} represent its maximum and minimum clipping factors, respectively. Subsequently, we use the Eq. (1) to quantize $\mathbf{W}_i$ and then calculate the quantization error using the MSE distance:

$$\mathcal{L}^i = \frac{1}{n}\left\|\mathbf{W}_i - \hat{\mathbf{W}}_i\right\|_2^2. \tag{18}$$

The final step involves adjusting the optimal maximum or minimum clipping factor of the quantization interval if the quantization error in the current round is less than that in the previous round.

5 Experiments

5.1 Settings

Models and Datasets. We evaluate MPPQ on LLaMA-1 (7B, 13B), LLaMA-2 (7B, 13B), and OPT[2] (125M, 1.3B, 2.7B, 6.7B, 13B) models. In zero-shot question-answering (QA) tasks, we report the accuracy of models on several datasets: PIQA [Bisk *et al.*, 2020], ARC [Clark *et al.*, 2018], HellaSwag [Clark *et al.*, 2018], WinoGrande [Sakaguchi *et al.*, 2021], and BoolQ [Clark *et al.*, 2019]. For language generation tasks, we measure the perplexity (PPL) on datasets including WikiText2 [Merity *et al.*, 2016] and C4 [Raffel *et al.*, 2020].

Quantization Settings. In terms of the quantization granularity, we employ per-token and per-channel quantization for activation and weight, respectively. Regarding the objects of quantization, we perform weight-only and weight-activation quantization. Meanwhile, the attention probabilities are maintained in full-precision for alignment with prior research. We also focus on quantization at low-bitwidth, including W2A16 and W4A4, among others.

Baseline Methods. Several recently prominent methods have been selected for benchmarking against MPPQ to highlight its superiority, including GPTQ [Frantar *et al.*, 2022], AWQ [Lin *et al.*, 2024], OmniQuant [Shao *et al.*, 2024], AffineQuant [Ma *et al.*, 2024], and ABQ-LLM [Zeng *et al.*, 2024] for weight-only quantization and SmoothQuant [Xiao *et al.*, 2023], OmniQuant, QLLM [Liu *et al.*, 2024a], ABQ-LLM, and LRQuant [Zhao *et al.*, 2024] for weight-activation quantization.

Implementation Details. We randomly sample 128 segments with a sequence length of 2048 from WikiText2 as our calibration training dataset. During the training phase, we adopt the AdamW optimizer with zero weight decay. Each transformer block is employed for 20 epochs with a batch size of 1 to complete the quantization reconstruction. Besides, the learning rates for the learnable outlier transformation factors and weight clipping factors are set to 5e-3 and 5e-4, respectively. For the learnable matrices in the PAR, we set the rank r to 128 and the learning rate to 1e-4. The entire training procedure is completed on a NVIDIA RTX 6000 48G GPU and we use the LM Evaluation Harness Toolbox for evaluation.

5.2 Experimental Results

Experiments on Language Generation Tasks. Table 1 presents a comparison of the PPL results for LLaMA-1 and LLaMA-2 models. As can be seen, with the reduction in quantization bitwidth or the model size becomes smaller, the enhancements shown by MPPQ become more pronounced. In particular, in the W6A6 quantization of the LLaMA-2-7B model, the margin of PPL between the MPPQ and FP16 baseline has narrowed to 0.14 and 0.23, respectively, which underscores the effectiveness of our approach. Furthermore, MPPQ reduces the PPL of the LLaMA-7B model on the WikiText2 dataset from 11.48 (ABQ-LLM) to 8.91 under the W2A16 quantization configuration, which was previously unattainable. These consistent superiority at low-bitwidth settings

[2]All results about OPT models are reported in Appendix B.

Bits	Method	LLaMA-7B		LLaMA-13B		LLaMA-2-7B		LLaMA-2-13B	
		WikiText2	C4	WikiText2	C4	WikiText2	C4	WikiText2	C4
FP16	-	5.67	7.08	5.09	6.61	5.47	6.97	4.88	6.46
W6A6	SmoothQuant	6.03	7.47	5.42	6.97	6.20	7.76	5.18	6.76
	OmniQuant	5.96	7.43	5.28	6.84	5.87	7.48	5.14	6.74
	QLLM	5.89	7.34	5.28	6.82	5.72	7.31	5.08	6.71
	ABQ-LLM	5.81	7.27	5.21	6.77	5.63	7.21	5.00	6.64
	LRQuant	5.88	7.35	5.27	6.84	5.67	7.24	5.07	6.68
	MPPQ	**5.79**	**7.27**	**5.19**	**6.76**	**5.61**	**7.20**	**4.99**	**6.64**
W4A4	SmoothQuant	25.25	32.32	40.05	47.18	83.12	77.27	35.88	43.19
	OmniQuant	11.26	14.51	10.87	13.78	14.26	18.02	12.30	14.55
	QLLM	9.65	12.29	8.41	10.58	11.75	13.26	9.09	11.13
	ABQ-LLM	8.63	12.10	7.69	10.90	9.31	12.85	8.62	11.47
	LRQuant	11.25	14.14	11.26	13.19	12.75	15.82	12.23	14.02
	MPPQ	**8.42**	**11.23**	**7.57**	**10.07**	**8.85**	**11.87**	**8.13**	**10.66**
W2A16	GPTQ	2.1e3	689.13	5.5e3	2.5e3	7.7e3	NaN	2.1e3	323.12
	QmniQuant	15.47	24.89	13.21	18.31	37.37	90.64	17.21	26.76
	AffineQuant	9.53	14.89	7.54	12.46	35.07	572.22	12.42	23.67
	ABQ-LLM	11.48	15.74	9.34	12.28	13.11	17.81	13.09	20.49
	MPPQ	**8.91**	**11.94**	**7.39**	**9.79**	**9.40**	**12.91**	**7.53**	**10.08**
W3A16	GPTQ	8.06	9.49	6.76	8.16	8.37	9.81	6.44	8.02
	AWQ	11.88	13.26	7.45	9.13	24.00	23.85	10.45	13.07
	QmniQuant	6.49	8.19	5.68	7.32	6.58	8.65	5.58	7.44
	AffineQuant	6.30	8.03	5.60	7.20	6.55	8.57	5.62	7.56
	ABQ-LLM	6.29	8.01	5.56	7.24	6.28	8.10	5.44	7.26
	MPPQ	**6.14**	**7.84**	**5.46**	**7.12**	**6.02**	**7.84**	**5.29**	**7.06**

Table 1: Perplexity (↓) results comparison of LLaMA-1 and LLaMA-2 models under weight-only quantization and weight-activation quantization. "NaN" represents an infinity error.

highlights our proficiency in maintaining generative capability under aggressive compression rates, without being dependent on any quantization-specific.

Experiments on Zero-shot QA Tasks. We further evaluate the performance of MPPQ on multiple zero-shot benchmarks using the accuracy metric. As illustrated in Figure 3, on almost all datasets and models, MPPQ has significantly outperformed existing SOTA baseline methods, showcasing its impressive and powerful generalization ability. For instance, in Figure 3a, our average accuracy outperforms LRQuant by approximately 4% while for larger models (e.g., LLaMA-2-13B), this advantage still remarkable exists.

Combining the consistent gains elaborated in the above two subsections, we deduce that our MPPQ has established a new SOTA in any quantization scenario.

5.3 Ablation Study

We conduct detailed ablation study to thoroughly analyze the efficacy of the proposed MMS, PAR, and FCP. The experimental results in Table 2 and Figure 4 indicate the contributions of each component.

It can be observed from Table 2 that enabling FCP reduces the perplexity from 5.97 to 5.82 (0.15 ↓), emphasizing the critical importance of properly aligning quantization and clipping patterns, an aspect often overlooked in prior work. Building on this, the inclusion of PAR further improves accuracy by 0.78 under W3A16 as shown in Figure 4, demonstrating that adaptive rounding is one of the keys to significantly enhancing quantization quality. Finally, the combination version delivers the best overall performance and is selected as our default configuration.

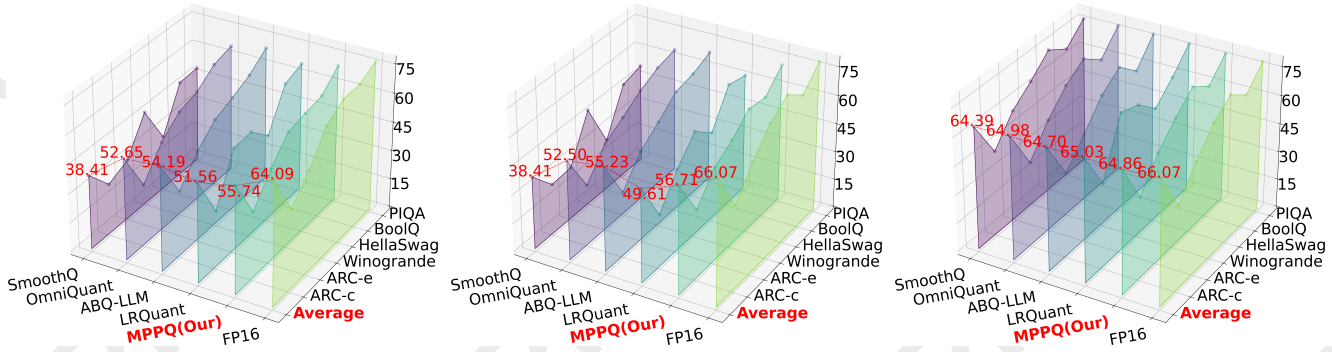
5.4 Discussions

The Effectiveness of MMS. Figure 2 indicates that our proposed MMS mitigates the inconsistency between distributions, while its ability to smooth outliers is shown in Figure 5.

It is evident that the Vanilla reflects the original distribution of activation, which is highly lacking in smoothness and is thus not preferred for quantization. The SmoothQuant and OmniQuant are able to yield relative improvements, yet both still have distinct limitations. In contrast, MPPQ provides

#Bits	Parts			PPL ↓	
	FCP	PAR	MMS	WikiText2	C4
W4A8				5.97	7.63
	✓			5.82	7.40
	✓	✓		5.66	7.26
	✓		✓	5.72	7.28
	✓	✓	✓	5.61	7.20
W3A16				6.63	8.64
	✓			6.54	8.56
	✓	✓		6.22	8.14
	✓		✓	6.45	8.32
	✓	✓	✓	6.02	7.84

Table 2: Ablation study on the primary innovations of MPPQ, including the FCP, PAR and MMS. All experiments are conducted on the LLaMA-2-7B model.



(a) LLaMA-1-7B_W4A4 (b) LLaMA-2-13B_W4A4 (c) LLaMA-2-13B_W6A6

Figure 3: Zero-shot QA (↑) accuracy of LLaMA-1 and LLaMA-2 models under weight-activation quantization.

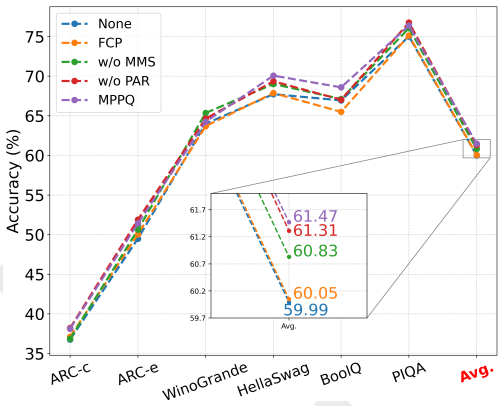
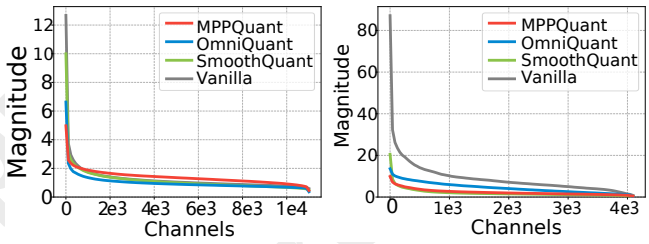
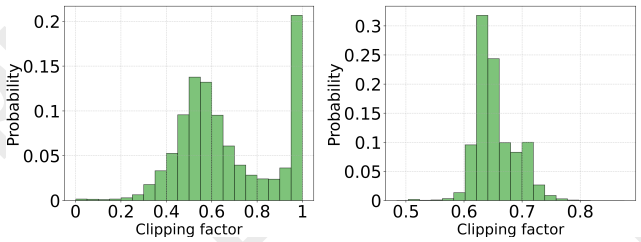


Figure 4: Ablation study on the contribution of different components to the improvement in accuracy.



(a) X_d of the 5^{th} transformer block. (b) X_o of the 31^{th} transformer block.

Figure 5: Visualization of the input distribution for linear layers down_proj and o_proj within certain transformer blocks of the LLaMA-2-7B model, arranged by the channel magnitudes (i.e., the Frobenius norm) in descending order.



(a) OmniQuant (b) MPPQ

Figure 6: Visualization of the learned clipping factors in W3A16 quantization of the LLaMA-2-7B model. Here, for a fair comparison, our MPPQ activates only the FCP component.

more precise smoothing for outliers and thus obtains betters gains in Table 1 and Figure 3.

Effectiveness Analysis of FCP. As depicted in Figure 6a, there are over 20% learned factors close to 1, indicating that numerous outliers still exist, which further leads to suboptimal quantization results. In contrast, our clipping factors are mostly concentrated between 0.6 and 0.75, implying effective clipping of outliers.

6 Conclusion

In this study, we have introduced MPPQ, a novel PTQ framework for LLMs that targets extremely low-bitwidth scenarios. MPPQ relies on scaling transformation for outlier handling and consists of three innovative optimizations. Firstly, MPPQ introduces a mixed metric supervision approach for block-wise quantization reconstruction to effectively alleviate the deviation between quantization and full-precision distributions, while providing better regularization guidance for parameter learning. Subsequently, we have successively proposes a proxy-based adaptive rounding scheme and a factor coarse pre-searching mechanism for weight quantization. The primary objective of the former is to minimize errors caused by the RTN operator, while the latter aims to ensure proper alignment between weight quantization and clipping patterns. Both can play a role in tackling their respective challenges. Extensive experimental results in language generation and zero-shot QA tasks indicate the superiority of the proposed method. For example, it reduces the gap in PPL to less than 0.5 with FP16 for W3A16 quantization on the LLaMA-7B model. Moreover, the models quantized by MPPQ, without any additional structures, facilitate their subsequent deployment on edge devices.

Ethical Statement

Our proposed method does not further amplify biases and contravene any ethical standards.

Acknowledgments

This work was supported by Beijing Natural Science Foundation under Grant L244050.

References

[Bengio *et al.*, 2013] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[Bisk *et al.*, 2020] 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*, volume 34, pages 7432–7439, 2020.

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

[Cheng *et al.*, 2023] Wenhua Cheng, Weiwei Zhang, Haihao Shen, Yiyang Cai, Xin He, Kaokao Lv, and Yi Liu. Optimize weight rounding via signed gradient descent for the quantization of llms. *arXiv preprint arXiv:2309.05516*, 2023.

[Choukroun *et al.*, 2019] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. Low-bit quantization of neural networks for efficient inference. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3009–3018. IEEE, 2019.

[Clark *et al.*, 2018] 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.

[Clark *et al.*, 2019] 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.

[Dettmers *et al.*, 2022] 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.

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

[Gong *et al.*, 2024] Ruihao Gong, Yang Yong, Shiqiao Gu, Yushi Huang, Chengtao Lv, Yunchen Zhang, Dacheng Tao, and Xianglong Liu. Llmc: Benchmarking large language model quantization with a versatile compression toolkit. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 132–152, 2024.

[Han *et al.*, 2015] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[Hu *et al.*, 2022] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022*, 2022.

[Kim *et al.*, 2024] Jeonghoon Kim, Jung Hyun Lee, Sungdong Kim, Joonsuk Park, Kang Min Yoo, Se Jung Kwon, and Dongsoo Lee. Memory-efficient fine-tuning of compressed large language models via sub-4-bit integer quantization. *Advances in Neural Information Processing Systems*, 36, 2024.

[Kullback and Leibler, 1951] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[Lee *et al.*, 2023a] Changhun Lee, Jungyu Jin, Taesu Kim, Hyungjun Kim, and Eunhyeok Park. Owq: Lessons learned from activation outliers for weight quantization in large language models. *arXiv preprint arXiv:2306.02272*, 2, 2023.

[Lee *et al.*, 2023b] 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*, pages 18913–18939. PMLR, 2023.

[Lee *et al.*, 2024] Jung Hyun Lee, Jeonghoon Kim, June Yong Yang, Se Jung Kwon, Eunho Yang, Kang Min Yoo, and Dongsoo Lee. Lrq: Optimizing post-training quantization for large language models by learning low-rank weight-scaling matrices. *arXiv preprint arXiv:2407.11534*, 2024.

[Li *et al.*, 2021] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: pushing the limit of post-training quantization by block reconstruction. In *9th International Conference on Learning Representations, ICLR 2021*, 2021.

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

[Liu *et al.*, 2023] Jiawei Liu, Lin Niu, Zhihang Yuan, Dawei Yang, Xinggang Wang, and Wenyu Liu. Pd-quant: Post-training quantization based on prediction difference metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24427–24437, 2023.

[Liu *et al.*, 2024a] 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 *The Twelfth International Conference on Learning Representations, ICLR 2024*, 2024.

[Liu *et al.*, 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 *Findings of the Association for Computational Linguistics, ACL 2024*, pages 467–484, 2024.

[Ma *et al.*, 2024] 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, ICLR 2024*, 2024.

[Merity *et al.*, 2016] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[Nagel *et al.*, 2020] Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119, pages 7197–7206, 2020.

[OpenAI, 2023] R OpenAI. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2(5), 2023.

[Raffel *et al.*, 2020] 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.

[Sakaguchi *et al.*, 2021] 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.

[Shao *et al.*, 2024] 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, ICLR 2024*, 2024.

[Touvron *et al.*, 2023a] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[Touvron *et al.*, 2023b] 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.

[Wei *et al.*, 2022a] Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. Qdrop: Randomly dropping quantization for extremely low-bit post-training quantization. In *The Tenth International Conference on Learning Representations, ICLR 2022*, 2022.

[Wei *et al.*, 2022b] 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.

[Wei *et al.*, 2023] 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.

[Xiao *et al.*, 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*, pages 38087–38099. PMLR, 2023.

[Yao *et al.*, 2022] 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.

[Yuan *et al.*, 2023] 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. *arXiv preprint arXiv:2304.01089*, 2023.

[Zeng *et al.*, 2024] 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. *arXiv preprint arXiv:2408.08554*, 2024.

[Zhao *et al.*, 2024] Jiaqi Zhao, Miao Zhang, Chao Zeng, Ming Wang, Xuebo Liu, and Liqiang Nie. Lrquant: Learnable and robust post-training quantization for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2240–2255, 2024.