# DL-QAT: Weight-Decomposed Low-Rank Quantization-Aware Training for Large Language Models

**Wenjin Ke, Zhe Li, Dong Li, Lu Tian, Emad Barsoum**
Advanced Micro Devices, Inc., Beijing, China
{wenjing.ke, z.li, d.li, lu.tian, emad.barsoum}@amd.com

## Abstract

Improving the efficiency of inference in Large Language Models (LLMs) is a critical area of research. Post-training Quantization (PTQ) is a popular technique, but it often faces challenges at low-bit levels, particularly in downstream tasks. Quantization-aware Training (QAT) can alleviate this problem, but it requires significantly more computational resources. To tackle this, we introduced Weight-**D**ecomposed **L**ow-Rank **Q**uantization-**A**ware **T**raining (DL-QAT), which merges the advantages of QAT while training only less than $1\%$ of the total parameters. Specifically, we introduce a group-specific quantization magnitude to adjust the overall scale of each quantization group. Within each quantization group, we use LoRA matrices to update the weight size and direction in the quantization space. We validated the effectiveness of our method on the LLaMA and LLaMA2 model families. The results show significant improvements over our baseline method across different quantization granularities. For instance, for LLaMA-7B, our approach outperforms the previous state-of-the-art method by $4.2\%$ in MMLU on 3-bit LLaMA-7B model. Additionally, our quantization results on pre-trained models also surpass previous QAT methods, demonstrating the superior performance and efficiency of our approach.

## 1 Introduction

Large language models (LLMs) have demonstrated exceptional performance across a variety of natural language processing (NLP) tasks. With the growing deployment and use of these models, quantization has become an essential method for reducing memory usage and enhancing computational efficiency. In LLM compression, a range of post-training quantization (PTQ) techniques have been developed, such as weight-only and weight-activation quantization. These techniques generally use a small calibration dataset and apply learning or optimization strategies to quickly transform a pre-trained floating-point model into a quantized version. However, PTQ methods struggle in low-bit quantization, especially in the downstream tasks. Despite the potential benefits, the development of quantization-aware training (QAT) algorithms has been constrained. This is primarily due to the significant data and computational resources required for comprehensive model fine-tuning, making it a costly endeavor.

To address the high computational expense associated with training LLMs, the Parameter-Efficient Fine-Tuning (PEFT) methodology has been introduced. PEFT entails fine-tuning only a fraction of the model's parameters, as opposed to the entirety, thereby enabling the efficient adaptation of pre-trained models to a diverse range of downstream applications. Notably, the Low-Rank Adaptation (LoRA) (Hu et al., 2021) technique, which represents the current state-of-the-art in PEFT, has been shown to achieve performance on par with fully fine-tuned models across various downstream tasks, without necessitating alterations to the model's inference architecture. The conventional approach to generating a quantized model for downstream tasks involves a two-step process: first, the floating-point model is fine-tuned on the downstream tasks; second, PTQ is applied to the fine-tuned model. However, this methodology is not without its drawbacks, as it can be cumbersome and may result in a substantial loss of accuracy. Conversely, directly employing QAT methods can lead to prohibitively high computational costs due to the requirement of end-to-end fine-tuning of all the model's parameters. The objective of our research is to devise a seamless, end-to-end process that yields a quantized model with parameter-efficient fine-tuning, thereby mitigating the aforementioned challenges and enhancing the overall efficiency and effectiveness of model adaptation for downstream tasks.



*Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 113–119
November 12-16, 2024 

Building upon these considerations, we propose Weight-**D**ecomposed **L**ow-Rank **Q**uantization-**A**ware **T**raining (DL-QAT), a novel end-to-end method designed to enhance the efficiency and effectiveness of model quantization for downstream tasks. DL-QAT decomposes the optimization of quantized weights into two processes: group-specific magnitude training and weight fine-tuning within a predefined quantization space. By incorporating a magnitude term, we calibrate the overall scale for each quantization group, ensuring a more precise representation of the model's parameters. Furthermore, we leverage low-rank matrices $A$ and $B$ to refine the quantized weights, thereby enhancing the model's adaptability to the specific requirements of the downstream tasks. To validate the efficacy of our approach, we conducted comprehensive experiments on the LLaMA and LLaMA2 model families. The results demonstrate a significant improvement over the baseline method, QA-LoRA (Xu et al., 2023), across various quantization granularities. Specifically, our method surpasses QA-LoRA by $+4.2\%$ on the MMLU benchmark (Hendrycks et al., 2020) and by $+5.5\%$ on the LM-Eval benchmark (Gao et al., 2023). Additionally, when compared to the previous state-of-the-art LLM-QAT method (Liu et al., 2023), our approach achieves lower perplexity on the WikiText-2 dataset (Merity et al., 2016) and higher accuracy on the LM-Eval benchmark, underscoring the superior performance of DL-QAT. LLM-QAT requires fine-tuning the entire model parameters, while we only need to fine-tune less than $1\%$ of the parameters to achieve better results. These findings not only highlight the effectiveness of DL-QAT in achieving competitive accuracy levels but also emphasize its efficiency in terms of both parameters and memory usage. By requiring minimal parameter modifications, DL-QAT offers a compelling alternative to traditional quantization methods, particularly for scenarios where computational resources are limited or where the need for rapid model adaptation is paramount.

## 2 Related work

**Parameter-Efficient Fine-Tuning.** LoRA (Low-Rank Adaptation) is a key method in Parameter-Efficient Fine-Tuning (PEFT), training a small number of parameters without altering the model inference process. To enhance its capabilities, variants like AdaLoRA(Zhang et al., 2023) and Pissa(Meng et al., 2024a) enhance rank via Singular Value Decomposition (SVD), while PLoRA(Meng et al., 2024b) accumulates low-rank updates progressively. Further, studies like (Zhu et al., 2024) and LoRA+ (Hayou et al., 2024) delve into the update mechanisms of LoRA's $A$ and $B$ matrices. DoRA (Liu et al., 2024) proposed a new optimization approach for LoRA, which decomposes LoRA updates into separate magnitude and direction updates to improve accuracy. Inspired by this idea, we further decompose LoRA quantization-aware training into fine-tuning the magnitude for quantization groups and fine-tuning the weights within the quantization space.

**Quantization of LLM.** Quantization has been widely used in LLM. Based on whether training is required, quantization can be classified into Post-Training Quantization(PTQ) and Quantization-Aware Training(QAT). PTQ methods requires only a small amount of calibration data to update the quantized weights. For instance, GPTQ (Frantar et al., 2022) utilizes merely 128 data samples to approximate second-order information and achieve the quantized weight. As outliers are crucial for LLM, considerable research is dedicated to addressing outlier issues. SmoothQuant (Xiao et al., 2023) effectively shifts the quantization challenge from activations to weights through a mathematically equivalent transformation. QuaRot (Ashkboos et al., 2024) employs Hadamard transformations on the weight matrices and attention modules to mitigate outlier effects. Compared with PTQ methods, QAT methods require more training data and resources, but generally achieve better performance. LLM-QAT (Liu et al., 2023) leverages data generated by pre-trained LLMs and achieves better performance compared with GPTQ. However, LLM-QAT requires significant training resources.

**Methods combining LoRA and quantization.** Building upon LoRA, QLoRA(Dettmers et al., 2024) was the first to propose a memory-efficient fine-tuning method by quantizing the pretrained model to low-bit and fine-tuning a high-precision LoRA component. This approach enables effective fine-tuning of LLMs within limited memory resources. Subsequent methods such as LoFTQ (Li et al., 2023) and LQ-LoRA (Guo et al., 2023) further optimized the initialization of the LoRA component and reduced the memory required for the quantized pretrained model. However, the combination of a low-bit pretrained model and a high-precision LoRA component still resulted in a high-precision weight after merging, which did not improve inference speed. To address this issue, QA-LoRA(Xu et al., 2023) made further improvements on QLoRA by learning an additional high-precision group-wise bias for the quantized model, effectively reducing both time and memory consumption without compromising accuracy. However, QA-LoRA could only perform group-wise fine-tuning, resulting in significant accuracy degradation when the quantization granularity increased.

## 3 Methodology

### 3.1 Low-Rank Adaptation and Quantization

In large language models (LLMs), a linear layer is denoted by $Y = W \cdot X$, where $W$ represents the weight matrix with dimensions $\mathbb{R}^{C_{out} \times C_{in}}$ and $X$ is the input with dimensions $\mathbb{R}^{C_{in} \times T}$. Here, $C_{out}$ and $C_{in}$ denote the output channel and input channel, respectively, and $T$ represents the sequence length. LoRA (Low-Rank Adaptation) refines the model by introducing two low-rank matrices, $A$ and $B$, where $A \in \mathbb{R}^{r \times C_{in}}$ and $B \in \mathbb{R}^{C_{out} \times r}$, with $r$ being the rank of LoRA matrix and $r \ll C_{in}, C_{out}$. The weight matrix $W$ is then modified as:

$$W = W_0 + \alpha BA \tag{1}$$

where $W_0$ represents the pretrained weight matrix that remains frozen during training, and $\alpha$ is a scaling factor that adjusts the influence of the low-rank adaptation.

For a given bit level $n$, the asymmetric weight quantization and dequantization processes can be described by a specific formula:

$$\tilde{w} = clip\left(\left\lfloor \frac{W - b}{s} \right\rceil, -2^{n-1}, 2^{n-1} - 1\right) \tag{2}$$

$$W_q = s * \tilde{w} + b \tag{3}$$

where $\tilde{w}$ represents the quantized value, while $W$ is the original floating-point weight. The scale $s$ determines the step size between quantization levels, and $b$ is the offset applied to the weight before scaling. The round function is denoted by $\lfloor \cdot \rceil$, and the $clip$ function ensures that the quantized values stay within the range $(-2^{n-1}, 2^{n-1} - 1)$. Dequantization involves converting the quantized values back to floating-point weights by scaling the quantized value with $s$ and adding the offset $b$, thus retrieving the original weight.

Quantization-Aware Training (QAT) simulates quantization during the forward pass by substituting $W$ with $W_q$, as depicted in equations 2 and 3, and employs the Straight-Through Estimator (STE) for gradient backpropagation to achieve the quantization effect. In LoRA, rather than updating the weight matrix $W$ directly, the updates are applied to the LoRA matrices $A$ and $B$. As a result, the quantization and de-quantization formula is modified accordingly:

$$\tilde{w'} = \text{clip}\left(\left\lfloor \frac{W_0 + \alpha BA - b}{s} \right\rceil, -2^{n-1}, 2^{n-1} - 1\right) \tag{4}$$

$$W_q' = s * \tilde{w'} + b \tag{5}$$

These formulas guarantee the integration of quantization effects into the LoRA weight updates, enabling efficient and precise training with quantization.

### 3.2 Weight-Decomposed Quantization

Rather than directly substituting $W$ with $W_q'$ in the quantization formula as indicated in equation 5 for QAT, or updating the $A$ and $B$ matrices along with the quantization parameters $s$ and $b$, we separate the joint training of LoRA and quantization into two parts: (1) group-specific magnitude training; (2) weight fine-tuning in the pre-defined quantization space. The quantization process is thus reformulated as follows:

$$\begin{aligned} W_q &= m * W_q' \\ &= m * (W_0 + \alpha BA)_q \\ &= m * (s * (\widetilde{W_0 + \alpha BA}) + b) \end{aligned} \tag{6}$$

Here, $m$ represents a newly introduced hyper-parameter denoting the group-specific magnitude, which matches the number of quantization groups and is identical in size to $s$. The matrix $m$ is initialized as a matrix of all ones. LoRA matrix $A$ is initialized with a random Gaussian distribution, and $B$ is initialized as a zero matrix. The variables $s$ and $b$ are initialized to map the range $(Min(W_0), Max(W_0))$ to the endpoints of the quantization interval. Therefore, $s_{\text{init}} = \frac{Max - Min}{2^n - 1}$, and $b_{\text{init}} = \frac{2^{(n-1)} \cdot Max + (2^{(n-1)} - 1) \cdot Min}{2^n - 1}$.

During the initial training phase, the scale factors $s$ and the biases $b$ are trained to ensure that the quantization updates commence from a well-established quantization space. Specifically, updates are applied only to $s$ and $b$ to obtain their initial values $s_0$ and $b_0$, which are then frozen. Subsequent training involves parameter optimization in two parts: **group-specific magnitude training** and **weight finetuning within the predefined quantization space**. The first part involves adjusting the magnitude term $m$ to set the scale for each quantization group, while in the second part, the $A$ and $B$ matrices are fine-tuned, permitting updates to the quantized weights within the established quantization space.

Our proposed method, DL-QAT, ensures a harmonious balance between the constraints imposed by quantization and the optimization of weights to achieve optimal model performance. By integrating the efficient fine-tuning capabilities of LoRA, DL-QAT not only streamlines the training process but also significantly reduces the associated computational costs and resource expenditure. This synergistic approach allows for the realization of state-of-the-art results while maintaining a high degree of efficiency, making it a compelling choice for scenarios where both performance and resource constraints are of paramount importance.

## 4 Experiments

In this section, we assess our approach using both language generation and zero-shot few-shot tasks with open-source models LLaMA-7B/13B (Touvron et al., 2023a) and LLaMA2-7B/13B (Touvron et al., 2023b) to demonstrate its effectiveness.

### 4.1 Experiment Setup

**Dataset.** We use Stanford-Alpaca dataset (Taori et al., 2023) as the fine-tuning dataset. Alpaca comprises a dataset of 52,000 instructions and demonstrations created by OpenAI's text-davinci-003 engine. This instructional data can be utilized to perform instruction-tuning on language models, enhancing their ability to follow instructions more effectively.

**Training Details.** In all experiments, a batch size of 16 was maintained, and a constant learning rate of 2e-4 was used. The optimizer employed was `adamw_hf`, with the default LoRA rank set at 16. For consistency with QALoRA's settings, training was conducted for 10,000 iterations, while other experimental results underwent 5,000 iterations. The training iterations for learning $s_0$ and $b_0$ were uniformly set at 1000. This approach ensures fair comparisons and reliable results across various models and datasets. Our experimental setup involves a quantization simulation in which all learnable parameters are represented in bf16 format. During inference, these quantized weights are dequantized back to bf16 for computation. We conducted all experiments on AMD MI-250 GPUs to maintain consistent hardware conditions.

**Evaluation Tasks.** The evaluation encompassed a broad spectrum of benchmarks. For language generation tasks, the perplexity on WikiText-2 (Merity et al., 2016) was reported. Additionally, results on the Massively Multitask Language Understanding (MMLU) benchmark (Hendrycks et al., 2020) were presented in both zero-shot and five-shot settings. The method was also assessed on seven common sense reasoning tasks from the EleutherAI LM Harness (Gao et al., 2023) for zero-shot performance.

### 4.2 Results

Our evaluation spanned various quantization granularities, including group-wise and channel-wise quantization. In group-wise quantization, we employed a standard setting with a group size of 128. For channel-wise quantization, our experiments encompassed two scenarios: one with quantization applied solely to weights, and another with quantization extended to weights, activations, and the kv cache.

Our approach was evaluated against prior quantization-aware LoRA-based methods, using QA-LoRA as the benchmark. To ensure a thorough comparison, we replicated the QA-LoRA algorithm with a group size of 128 and channel-wise quantization, while preserving its original LoRA rank of 64. The results presented in Table 1 and Table 2 demonstrate that our technique surpasses the benchmark across different quantization bits, granularities, and datasets. Remarkably, we noted a $+4.2\%$ enhancement in MMLU zero-shot accuracy on LLaMA-7B with 3-bit group-wise quantization, and a $+5.5\%$ increase in Common Sense QA accuracy on LLaMA2-7B with 4-bit per-channel quantization.

Moreover, we conducted comparisons with the PTQ method SmoothQuant (Xiao et al., 2023) and the QAT method LLM-QAT (Liu et al., 2023) on the LLaMA-7B/13B models within the W4A8KV8 framework, as depicted in Table 2. Our approach yielded lower perplexity scores compared to LLM-QAT. In terms of common sense QA accuracy, it

| LLaMA | Method | Bits | MMLU | | Common Sense Zero-Shot | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0-Shot | 5-Shot | ARC_C | ARC_E | BoolQ | HellaSwag | OBQA | PIQA | Winogrande | Avg |
| **1-7B** | - | 16 | 32.1 | 34.6 | 38.2 | 67.3 | 72.9 | 56.3 | 28.4 | 78.2 | 67.1 | 58.3 |
| | QA-LoRA* | 4 | 37.9 | 38.5 | 44.0 | 71.6 | 75.9 | 57.1 | 30.8 | 78.9 | 67.2 | 60.8 |
| | Ours | 4 | **40.5** | **39.9** | **45.0** | **75.5** | **79.8** | **57.9** | **36.2** | **78.9** | **70.2** | **63.4** |
| | QA-LoRA* | 3 | 32.2 | 32.9 | **41.7** | 71.6 | 76.9 | 54.6 | 28.0 | 77.6 | 64.9 | 59.3 |
| | Ours | 3 | **36.4** | **33.9** | 41.0 | **73.4** | **78.2** | **55.3** | **34.2** | **78.2** | **67.5** | **61.1** |
| **2-7B** | - | 16 | 40.7 | 45.5 | 39.9 | 69.3 | 71.1 | 56.7 | 31.8 | 78.3 | 67.1 | 59.2 |
| | QA-LoRA* | 4 | 42.5 | 44.8 | 42.7 | 71.9 | 77.6 | 56.9 | 32.6 | **79.2** | 68.3 | 61.3 |
| | Ours | 4 | **44.6** | **45.0** | **47.2** | **77.8** | **79.3** | **58.1** | **35.6** | 78.5 | **68.5** | **63.6** |
| | QA-LoRA* | 3 | 37.9 | 37.9 | 38.1 | 66.6 | 75.0 | 54.0 | 32.0 | 76.0 | 66.5 | 58.3 |
| | Ours | 3 | **40.5** | **39.4** | **41.2** | **74.4** | **78.0** | **54.7** | **32.2** | **77.5** | **68.8** | **60.9** |

Table 1: Results of weight-only group-wise quantization with group_size=128 on LLaMA-7B and LLaMA2-7B. The evaluation includes results for MMLU (both 0-shot and 5-shot settings) and Common Sense QA Zero-shot tasks ($acc$ is reported to maintain consistency with QA-LoRA). * indicates reproduced results.

| LLaMA | Method | W-A-KV | Wikitext2 | Common Sense Zero-Shot | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | ppl (↓) | ARC_C | ARC_E | BoolQ | HellaSwag | PIQA | Winogrande | Avg |
| 1-7B | - | 16-16-16 | 5.68 | 48.0 | 73.0 | 76.8 | 76.1 | 79.3 | 70.0 | 70.5 |
| | QA-LoRA* | 3-16-16 | 16.5 | 38.4 | 51.5 | 64.3 | 64.5 | 73.7 | 60.9 | 58.9 |
| | Ours | 3-16-16 | **9.2** | **40.1** | **61.8** | **71.2** | **67.2** | **75.9** | **64.0** | **63.4** |
| | QA-LoRA* | 4-16-16 | 11.1 | 42.4 | 58.0 | 73.7 | 70.5 | 77.3 | 66.1 | 64.7 |
| | LLM-QAT | 4-16-16 | - | **45.0** | **70.0** | 75.5 | 74.0 | **78.3** | **69.0** | 68.6 |
| | Ours | 4-16-16 | **6.7** | 44.4 | 68.5 | **78.5** | **74.4** | 78.1 | 68.5 | **68.7** |
| | SmoothQuant | 4-8-8 | - | 42.8 | 67.4 | 71.0 | 67.8 | 77.6 | 66.0 | 65.2 |
| | LLM-QAT | 4-8-8 | - | 45.6 | 70.2 | 74.6 | 73.5 | 77.5 | 67.7 | 68.2 |
| | Ours | 4-8-8 | **6.7** | **46.2** | **71.3** | **78.1** | **73.6** | **78.5** | **68.4** | **69.4** |
| 1-13B | - | 16-16-16 | 5.09 | 52.6 | 74.5 | 78.1 | 79.2 | 80.0 | 73.6 | 73.0 |
| | SmoothQuant | 4-8-8 | - | 43.3 | 67.4 | 72.5 | 74.3 | 77.1 | 69.5 | 67.4 |
| | LLM-QAT | 4-8-8 | - | **51.9** | 73.6 | 77.5 | 73.6 | 79.1 | **70.6** | 71.6 |
| | Ours | 4-8-8 | **5.9** | 48.8 | **74.8** | **80.5** | **77.1** | **80.4** | 70.3 | **72.0** |
| 2-7B | - | 16-16-16 | 5.47 | 46.3 | 74.6 | 77.7 | 76.0 | 79.1 | 69.1 | 70.5 |
| | QA-LoRA* | 3-16-16 | 13.7 | **36.3** | 48.2 | 70.3 | **66.3** | 74.4 | **63.9** | 59.9 |
| | Ours | 3-16-16 | **9.4** | 35.9 | **58.9** | **71.1** | 63.6 | **74.8** | 60.2 | **63.7** |
| | QA-LoRA* | 4-16-16 | 9.5 | 41.3 | 55.1 | 68.8 | 71.9 | 77.3 | 68.2 | 63.8 |
| | Ours | 4-16-16 | **6.3** | **44.6** | **71.0** | **78.5** | **74.6** | **78.2** | **68.8** | **69.3** |
| 2-13B | - | 16-16-16 | 4.88 | 49.0 | 77.4 | 80.6 | 79.4 | 80.5 | 72.2 | 73.2 |
| | Ours | 4-8-8 | 5.63 | 49.6 | 75.5 | 81.9 | 78.1 | 80.1 | 70.3 | 72.6 |

Table 2: Results of channel-wise quantization results on LLaMA-7B/13B and LLaMA2-7B/13B models. Evaluation metrics include perplexity (ppl) on WikiText-2 and accuracy in common sense QA zero-shot tasks. $Acc\_norm$ is reported to ensure consistency with LLM-QAT. * indicates reproduced results.

substantially surpasses SmoothQuant and LLM-QAT. Moreover, our approach necessitates significantly less training memory and time compared to LLM-QAT, proving that our DL-QAT method not only yields superior outcomes but also enhances efficiency.

### 4.3 Ablation Study

To demonstrate the effectiveness of our introduced group-specific magnitude $m$ and our quantization update strategy, including weight fine-tuning in the pre-defined quantization space, we conducted ablation experiments as shown in Table 3.

For quantization updates, we considered three possible settings: (1) Min-Max Clipping Values: Quantization values are uniformly distributed between the updated $\min(W_0+\alpha BA)$ and $\max(W_0+\alpha BA)$, with clipping always performed at these dynamic bounds. (2) Fixed Clipping Values: The clipping values are fixed by learned $s_0$ and $b_0$, ensuring that $W_0+\alpha BA$ updates within a fixed quantization space. (3) Adaptive Clipping Values: Both $s$ and $b$ are continuously trained, adaptively updating the quantization space throughout the training process. For the magnitude $m$, we explored two possible settings: with or without the learnable magnitude term $m$.

The results in Table 3 show that experiments with the learnable magnitude $m$ consistently outperform those without it. This indicates that using $m$ to adjust the quantization group's magnitude aids in adaptive scaling. Without the learnable

| Setting | m | Clipping bounds | Learnable params | LLaMA-7B 4 bit | 3 bit |
|---|---|---|---|---|---|
| 1 | N/A | MinMax | $A, B$ | 69.7 | 67.5 |
| 2 | N/A | Learn then fix | $s, b$ then $A, B$ | 70.4 | 66.7 |
| 3 | N/A | Learn | $s, b, A, B$ | 70.0 | 67.2 |
| 4 | Learn | MinMax | $m, A, B$ | 70.4 | 67.2 |
| 5 | Learn | Learn then fix | $s, b$ then $m, A, B$ | **70.7** | **68.3** |
| 6 | Learn | Learn | $m, s, b, A, B$ | 70.0 | 67.7 |

Table 3: Results with different magnitude and quantization settings on LLaMA-7B. Average $acc\_norm$ in common sense QA zero-shot tasks is reported. With a quantization granularity of group_size=128.

| LLaMA | Quant config | Trainable Params (M) $s, b$ | $m, A, B$ | GPU Memory (G) | Training speed (s/iter) |
|---|---|---|---|---|---|
| **7B** | Weight-only, g128 | 50 | 71 | 32.5 | 3.33 |
| | Weight-only, per-channel | 1 | 41 | 31.8 | 3.24 |
| | Quant W/A/KV, per-channel | 1 | 41 | 33.1 | 3.91 |
| **13B** | Weight-only, g128 | 99 | 162 | 60.4 | 6.26 |
| | Weight-only, per-channel | 2 | 65 | 58.7 | 6.09 |
| | Quant W/A/KV, per-channel | 2 | 65 | 62.8 | 7.04 |

Table 4: Training parameter count, GPU memory usage, and training speed for LLaMA-7B/13B under different quantization configurations with a per-GPU batch size of 16. The experiments were conducted on an AMD MI250 with 64GB of GPU memory.

magnitude $m$, accuracy across various bit settings varies, with no single setting being clearly superior. However, when combined with the learnable magnitude $m$, setting 2 — our proposed method of weight fine-tuning in the pre-defined quantization space — significantly outperforms the other settings. This suggests that our strategy of decomposing the weight into two parts for updates is effective, allowing the magnitude and weight distribution to be optimized separately, resulting in excellent fine-tuning outcomes.

### 4.4 Analysis

In Table 4, we evaluate the training parameter count, GPU memory usage, and training speed for LLaMA-7B and 13B models. The total parameters of LLaMA-7B and LLaMA-13B are 6.8G and 13.1G, respectively. For group-wise quantization, after fixing parameters $s$ and $b$, the remaining trainable parameters $m$ and $A, B$ account for only $1.0\%$ and $1.2\%$ of the total parameters in LLaMA-7B and LLaMA-13B, respectively. For channel-wise quantization, the training parameters constitute $0.6\%$ and $0.5\%$ of the total parameters for LLaMA-7B and LLaMA-13B, respectively. With a batch size of 16, our simulated quantized training shows that LLaMA 7B and 13B use a maximum of $33.1GB$ and $62.8GB$ of GPU memory, respectively. On the Alpaca dataset, with an AMD MI250 GPU, LLaMA-7B can train up to $17,669$ samples per hour, while LLaMA-13B can train up to $9,458$ samples per hour. Therefore, compared to the previous QAT methods, our approach takes only about one-thirtieth of the time to converge the model, significantly reducing the resources needed for training.

## 5 Conclusion

In this paper, we introduce Weight-**D**ecomposed **L**ow-Rank **Q**uantization-**A**ware **T**raining (DL-QAT), a novel end-to-end approach designed to improve the efficiency of QAT for tasks downstream of LLMs. DL-QAT optimizes quantized weights through two main processes: group-specific magnitude training and weight fine-tuning within a set quantization space. By employing Low-Rank Adaptation (LoRA) matrices, we are able to update the weight magnitude and direction within the quantization space, thereby enabling precise adjustments to the model's parameters. DL-QAT achieves remarkable results by training on less than $1\%$ of the model's parameters, outperforming previous QAT methods across established Natural Language Processing benchmarks. This efficiency in parameter utilization is a testament to the effectiveness of DL-QAT in achieving state-of-the-art performance while minimizing computational overhead.

## References


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

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

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

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. 2023. A framework for few-shot language model evaluation.

Han Guo, Philip Greengard, Eric P Xing, and Yoon Kim. 2023. Lq-lora: Low-rank plus quantized matrix decomposition for efficient language model finetuning. *arXiv preprint arXiv:2311.12023*.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. 2024. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, and Tuo Zhao. 2023. Loftq: Lora-fine-tuning-aware quantization for large language models. *arXiv preprint arXiv:2310.08659*.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2023. Llm-qat: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. 2024a. Pissa: Principal singular values and singular vectors adaptation of large language models. *arXiv preprint arXiv:2404.02948*.

Xiangdi Meng, Damai Dai, Weiyao Luo, Zhe Yang, Shaoxiang Wu, Xiaochen Wang, Peiyi Wang, Qingxiu Dong, Liang Chen, and Zhifang Sui. 2024b. Periodiclora: Breaking the low-rank bottleneck in lora optimization. *arXiv preprint arXiv:2402.16141*.

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

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

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

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

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

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhensu Chen, Xiaopeng Zhang, and Qi Tian. 2023. Qa-lora: Quantization-aware low-rank adaptation of large language models. *arXiv preprint arXiv:2309.14717*.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.

Jiacheng Zhu, Kristjan Greenewald, Kimia Nadjahi, Haitz Sáez de Ocáriz Borde, Rickard Brüel Gabrielsson, Leshem Choshen, Marzyeh Ghassemi, Mikhail Yurochkin, and Justin Solomon. 2024. Asymmetry in low-rank adapters of foundation models. *arXiv preprint arXiv:2402.16842*.