QER: Quantized Low-rank Error Reconstructor for LLM Low-bitwidth Quantization

Shoukai Xu
South China University of Technology
Guangzhou, China
shoukaixv@gmail.com

Runhao Zeng
SHENZHEN MSU-BIT UNIVERSITY
Shenzhen, China
zengrh@smbu.edu.cn

Zhiyang Zhang
South China University of Technology
Guangzhou, China
1963306815@qq.com

Hao Huang
South China University of Technology
Guangzhou, China
2632880737@qq.com

Qingfang Zheng
PengCheng Laboratory
Shenzhen, China
zhengqf01@pcl.ac.cn

Xiangyuan Lan
PengCheng Laboratory
Shenzhen, China
lanxy@pcl.ac.cn

Yaowei Wang
PengCheng Laboratory
Shenzhen, China
wangyw@pcl.ac.cn

Mingkui Tan
South China University of Technology
Guangzhou, China
mingkuitan@scut.edu.cn

2025 IEEE International Conference on Cloud Computing Technology and Science (CloudCom) | 979-8-3315-6634-0/25/$31.00 ©2025 IEEE | DOI: 10.1109/CLOUDCOM67567.2025.11331377

Abstract*—Large Language Models (LLMs) have achieved remarkable success but face significant deployment challenges in cloud and edge environments due to their massive computational and storage requirements. Model quantization serves as a key solution to enhance the scalability and efficiency of LLMs within distributed cloud platforms. Existing Post-Training Quantization (PTQ) methods often exhibit suboptimal performance in low-bit settings. To further improve their precision, Quantization-Aware Training (QAT) combined with Low-Rank Adaptation (LoRA) has been explored for error correction. However, a critical issue is that the quantized base model and full-precision LoRA parameters suffer from precision mismatch, introducing additional errors during weight merging. To address these challenges, we propose a Quantized Low-rank Error Reconstructor (QER) for LLM low-bitwidth quantization. QER first enables lossless merging in low-bitwidth format by aligning the bitwidth of its low-rank parameters with the quantized base parameters, eliminating dequantization and requantization steps. Through this process, QER reconstructs original errors into two components: the quantization errors of QER parameters (*i.e.*, quantized low-rank parameters) and potential overflow errors during low-bitwidth merging. These two errors are directly related to QER parameters, making them easier to optimize via gradient-based updates within an error-aware training framework. Requiring only 128 samples and 1 training epoch, QER demonstrates superior performance on LLaMA-1/2 families. In 4-bit quantization, compared to QLLM with error correction, QER reduces average perplexity by 13.8% (from 10.97 to 9.45) and improves average accuracy by 3.01 percentage points (from 51.84% to 54.85%) on LLaMA-1-7B. QER bridges the gap between quantization and low-rank adaptation, enabling efficient and accurate low-precision LLM deployment.

***Index Terms*—Large language model, efficient computing, quantization-aware training, error correction.**

I. Introduction

Large Language Models (LLM) have achieved great success [1]–[5], but this accomplishment has been accompanied by a large number of parameters and huge computational requirements. As a result, LLM inference needs significant computational resources and extensive GPU memory, which leads to high power consumption and sluggish inference speeds. These challenges are particularly critical in cloud and edge computing environments, where efficient resource utilization are essential for sustainable service delivery. To improve the scalability and efficiency of large language models when implementing LLM inference, model quantization is an important solution.

Quantization operations can be performed on both weights and activations. Activation quantization has a greater negative impact compared with weight quantization, which can inevitably lead to obvious performance degradation. In order to avoid severe performance degradation, two kinds of strategies have been exploited in early LLM quantization methods: one is to quantize the weight but not to quantize the activation [6], [7], and the other is to quantize both the weight and the activation value to 8-bit without considering lower bitwidth [8], [9]. However, if the activations are not quantified, the weights still need to be restored to FP16 for calculation in inference, which greatly reduces the calculation efficiency. Therefore, it is a challenging problem to realize the low bitwidth weight-

This work was partially supported by the Joint Funds of the National Natural Science Foundation of China (Grant No.U24A20327), TCL Science and Technology Innovation Fund, China, Shenzhen Science and Technology Foundation (No. JCYJ20250604173210013), Key Scientific Research Project of the Department of Education of Guangdong Province 2024ZDZX3012, Engineering Technology Research Center for Ordinary Universities in Guangdong Province (No.2024GCZX005), National Natural Science Foundation of China (62402252) and (62536003), Guangdong High-Level Talent Programme (2024TQ08X283).

Shoukai Xu, Runhao Zeng and Zhiyang Zhang are equal contributors.
Mingkui Tan, Yaowei Wang and Qingfang Zheng are corresponding authors.

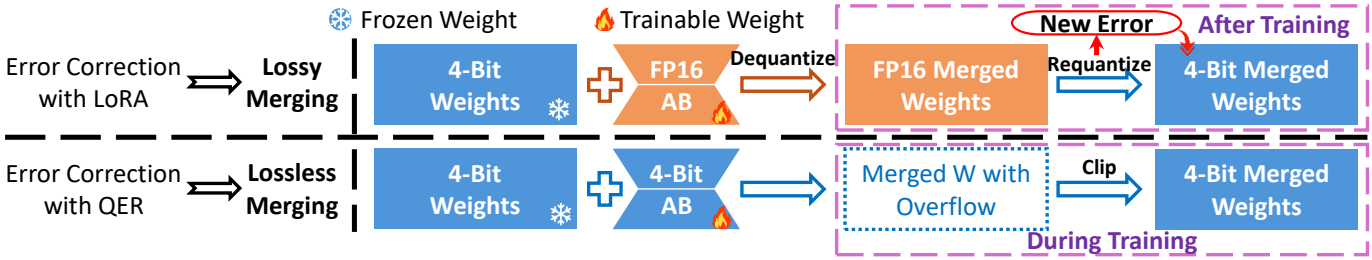

Fig. 1: Comparison of weight merging processes between traditional LoRA-based error correction and our QER. Traditional approach: Quantized base weights (4-bit) are dequantized to FP16, merged with full-precision (FP16/32) LoRA matrices (AB), and requantized to 4-bit. This lossy merging process introduces new quantization errors. Our QER: Both quantized base weights (4-bit) and quantized low-rank adapter weights $(AB)_q$ (4-bit) are merged directly in low-bitwidth format, avoiding dequantization and requantization. This enables lossless merging during training the weights $(AB)_q$.

activation quantization and ensure that the quantized LLM is within the acceptable performance degradation range.

Currently, some new works are tackling this problem. Post-training quantization (PTQ) methods [10], [11] address this by mitigating activation outliers, thereby reducing quantization errors and enhancing the performance of quantized models. The quantization-aware training (QAT) methods [12], [13] optimize model parameters through targeted training to achieve activation quantization with minimal accuracy loss. QAT essentially functions as an error correction mechanism, demonstrating significant capability in recovering model performance compromised by quantization processes. At present, performing error correction after post-training quantization has emerged as a notable trend [14]–[17]. This methodology has shown promising results in restoring model accuracy. Due to excessive memory and processing requirements, full-parameter fine-tuning of LLMs remains computationally prohibitive. Consequently, quantization with error correction frameworks frequently employ Low-Rank Adaptation (LoRA) for parameter updates, capitalizing on its efficiency advantages in training acceleration and resource conservation.

However, when applying LoRA for error correction in quantized LLMs, an additional new error is inevitably introduced, negatively impacting the final training outcomes. Specifically, the current implementation of LoRA uses full-precision (FP32) or half-precision (FP16) parameters, while the base model weights are quantized to low precision (e.g., INT4/INT8). When merging the quantized base weights with the full-precision LoRA adaptation matrices, the quantized weights must first be dequantized to full precision, and then the merging operation is performed in the floating-point domain. This results in merged weights in floating-point format. To produce a quantized model for deployment, these merged weights must undergo requantization, *i.e.*, an additional PTQ processing. The bitwidth mismatch between the quantized base model and the full-precision LoRA parameters introduces significant new errors during weight merging. This dequantization-merging-requantization process is illustrated in Figure 1.

To address this problem, we propose a Quantized Low-rank Error Reconstructor (QER) to facilitate the performance recovery for the quantized LLM. Firstly, our QER performs a lossless merging in low-bitwidth format. Specifically, we modify LoRA to operate directly in the quantized domain, ensuring that both the base model and low-rank adaptation parameters share the same bitwidth. The quantized low-rank adaptation parameters are called QER parameters. Then we merge the QER parameters into the base weights directly in the low bitwidth format. By aligning the bitwidth, we eliminate dequantization and requantization operations, thereby avoiding errors introduced during these processes. Through QER, we ensure lossless merging after error correction.

Secondly, we propose an error-aware training for QER weights. QER reconstructs the original dequantization and requantization errors into two components: quantization errors of QER parameters and potential overflow errors during low-bitwidth merging. These two errors are directly related to QER parameters, making them easier to optimize. Crucially, as fine-tuning parameters designed for error correction, QER parameters serve multiple purposes. During training, we freeze the weights of the base model and only train the quantized weights of the QER. Our contributions are as follows:

- This paper reveals the parameter bitwidth mismatch issue in the fusion of LLM quantization and low-rank adaptation. We propose a novel targeted Quantized Low-rank Error Reconstructor (QER) that operates natively in the quantized format to eliminate the bitwidth mismatch and reduce errors of LLM quantization.
- The proposed QER framework enables lossless merging in low-bitwidth formats by reconstructing the dequantization errors of base parameters and requantization errors of merged parameters into two addressable components: (1) quantization errors of low-rank parameters and (2) possible bitwidth overflow errors during merging. Crucially, both error types can be eliminated within our error-aware training process by directly updating the QER parameters. QER incorporates all errors into optimization objectives.
- Extensive experiments on LLaMA-1/2 demonstrate the superiority of our QER. For 4-bit LLaMA-1-7B, QER reduces average perplexity by 13.8% (9.45 vs. QLLM's 10.97) and improves zero-shot accuracy by 3.01 percentage points (54.85% vs. QLLM's 51.84%). In more challenging 3-bit quantization, QER is able to recover

catastrophic failure cases. Moreover, QER requires only 128 samples and 1 training epoch.

II. Related Work

A. Post-Training Quantization

Post-Training Quantization (PTQ) methods compress pre-trained LLMs without retraining. Existing PTQ methods can be divided into two primary categories: Weight-Only PTQ and Weight-Activation PTQ. Weight-Only PTQ only quantizes model weights, while activations remain in floating-point. GPTQ [6] employs Hessian-based error compensation to minimize layer-wise quantization errors. AWQ [7] optimizes weight scaling by preserving salient channels influenced by activation outliers. However, activations remain floating-point, limiting hardware acceleration.

Weight-Activation PTQ quantizes both weights and activations. The outlier problem of activations leads to accuracy loss. SmoothQuant [8] migrates quantization difficulty via mathematical transformations. OmniQuant [10] jointly optimizes quantization parameters and transformation coefficients. SpinQuant [18] leverages learned rotation matrices for 4-bit W4A4 quantization. DuQuant [19] introduces dual transformations (rotation and permutation) to redistribute activation outliers, achieving robust 4-bit performance. These methods address joint weight-activation quantization through distribution alignment and outlier mitigation. However, they often exhibit suboptimal performance in low-bit settings.

PTQ with Error Correction. To further enhance the performance of quantized LLMs, some PTQ methods integrate efficient error correction techniques after quantization. QLLM [15] applies low-rank adaptation to recover accuracy. ASER [14] leverages parameter-efficient fine-tuning (PEFT) for layer-wise error correction. However, the quantized weights must be dequantized back to FP16/32 format to properly integrate the full-precision LoRA modules. To obtain the final quantized model, another round of PTQ quantization is required on the error-corrected parameters. Unfortunately, this process reintroduces new quantization errors, undermining the effectiveness of error correction and ultimately degrading the performance of the quantized LLM.

B. Quantization-Aware Training

Quantization-Aware Training(QAT) incorporates quantization constraints during training for enhanced performance. Weight-only QAT [16], [17] adapts models to low-bit weights during training while maintaining near-lossless performance. Weight-Activation QAT, such as LLM-QAT [20] jointly optimizes model parameters by simultaneously accounting for both weight and activation quantization. Despite superior performance, it demands substantial computational resources and labeled datasets, limiting practicality.

III. Quantized Low-rank Error Reconstructor

A. Problem Definition

The inherent limitations of post-training quantization become particularly pronounced in ultra-low bitwidth cases ($\leq$4 bits), where performance degradation escalates non-linearly with quantization intensity. To mitigate these artifacts, quantization-aware training (QAT) emerges as a viable error compensation paradigm through fine-tuning parameters. However, full-parameter QAT proves computationally prohibitive for modern LLMs, as updating the entire parameter space demands huge memory and energy budgets. This constraint necessitates Low-Rank Adaptation (LoRA) as a grounded compromise. Its factorized parameterization reduces trainable parameters while preserving error correction capacity.

In this work, we uncover and analyze a critical yet overlooked conflict in the current integration strategies of Quantization-Aware Training (QAT) and Low-Rank Adaptation (LoRA) for large language models (LLMs). When employing LoRA for error correction in quantized LLMs, a bitwidth mismatch emerges between the quantized model backbone (typically implemented using INT4 or INT8 precision) and the full-precision LoRA parameters (commonly maintained in FP16 or FP32 format). This inconsistency introduces systemic errors during the critical weight merging phase, where the quantized weights are combined with the full-precision LoRA updates.

- **Precision Mismatch in Weight Merging**: Given quantized backbone weights $W_q \in \mathbb{Z}^{n\times m}$ with quantization parameters (s, z), and LoRA matrices $A \in \mathbb{R}^{n\times r}$, $B \in \mathbb{R}^{r\times m}$, the merging operation requires:

$$W_{\text{merged}} = \underbrace{s(W_q - z)}_{\text{dequantized}} + AB \tag{1}$$

  This forces the merged weights to revert to floating-point representation, as illustrated in Figure 1.
- **Requantization Loss**: The final deployment requires requantizing W_{merged}:

$$W_q' = \text{Quantize}(W_{\text{merged}}) \tag{2}$$

$$\mathcal{L}_{\text{requant}} = \|W_{\text{merged}} - s'(W_q' - z')\|_2^2 \tag{3}$$

  where $\mathcal{L}_{\text{requant}}$ accumulates the requantization error.

B. Initialization of the QER

The initialization process of the Quantized Low-Rank Error Reconstructor (QER) is important for its effectiveness. The initialization quality of QER significantly impacts model convergence speed and optimization performance. We start by leveraging the Post-Training Quantization (PTQ) method, such as QLLM [15], to quantize the parameters of the pre-trained model. This step provides the initial quantized weights W_q for the model. The quantization process can be mathematically represented as:

$$W_q = \text{Quantize}(W, s, z), \tag{4}$$

where W are the original pre-trained weights, s is the scale factor, and z is the zero-point.

In model quantization, precision reduction from full-precision to low-bit representations inevitably induces numerical discrepancies due to the inherent information bottleneck

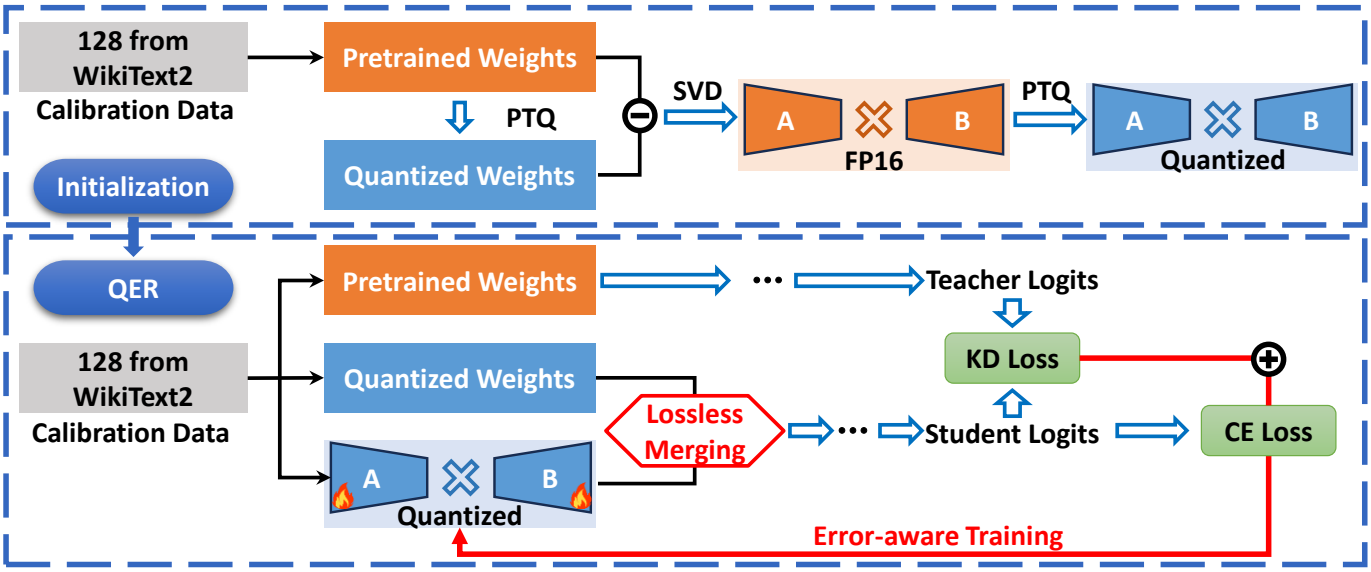

Fig. 2: The framework of the Quantized Low-rank Error Reconstructor (QER) for LLM low-bitwidth quantization. The process starts with pretrained FP16 weights, which are first quantized using Post-Training Quantization to obtain initial quantized weights. The difference between the pretrained full-precision weights and quantized weights (ΔW) is decomposed via SVD into low-rank matrices A and B, which are then quantized to form the initial QER weights. Error-aware training using only 128 calibration samples from WikiText2. The quantized base weights and quantized QER weights $(AB)_q$ are merged in low-bitwidth format, and the training optimizes QER weights using both cross-entropy (CE) and knowledge distillation (KD) loss.

in low-bitwidth numerical formats. These errors can be formalized as:

$$\Delta W = W - W_q. \tag{5}$$

The primary objective of error recovery is to refine quantized parameters through targeted adjustments, thereby minimizing output deviations from the original full-precision model.

To address this quantization-induced error, we propose an error recovery mechanism that refines quantized parameters through trainable error correction weights W_{ec}. We construct composite weights of the quantized LLM via $W_{merged} = W_q + W_{ec}$. The fundamental optimization target requires $W_{merged} = W$. This leads to the $W_{ec} = W - W_q = \Delta W$, ensuring the merged weights asymptotically approximate the original full-precision parameters. Thus, we use the ΔW as the initialization of the W_{ec}.

Then, we construct the low-rank format of the error corrector. We perform a low-rank decomposition on W_{ec}. Singular Value Decomposition (SVD) is used to decompose W_{ec} into two matrices A and B such that $W_{ec} \approx AB$. This low-rank representation significantly reduces the number of parameters that need to be trained.

C. Lossless Merging in Low-bitwidth Format

For full-precision models, after the LoRA branch updates are completed, the parameters of the LoRA branch need to be merged into the base parameters to ensure no additional computational overhead during the final model inference. The merging of the base parameters and LoRA parameters is a simple addition process and is lossless. However, for quantized models, issues arise during merging. The mismatch in bitwidth between the two introduces additional loss to the final model. The base parameters of the model are quantized into lower bitwidths, while the directly used LoRA is full-precision.

Therefore, our Quantization Low-rank Error Reconstructor (QER) aims to resolve the bitwidth mismatch issue, as shown in Figure 2. Unlike existing methods that directly use LoRA, we choose to design and construct a low-bitwidth error reconstructor for quantized LLM models, ensuring it operates in the same low-bitwidth domain as the base parameters. This allows direct merging in the low-bitwidth format, avoiding the introduction of new errors from additional requantization. Specifically, we use the same quantizer as the base parameters to quantize the low-rank matrices. We quantize their multiplication (AB) to get our QER (quantized error reconstructor) parameters $(AB)_q$. The merging of the quantized base weights and the low-rank QER weights is defined as:

$$W_{\text{merged}} = W_q + (AB)_q. \tag{6}$$

Since the low-rank parameters to be merged with the base parameters are the product of (AB), using multiplication is well-suited for merging. We do not separately quantize A and B before multiplying them for two reasons: first, it reduces the number of quantizations, minimizing quantization-induced errors; second, it avoids bitwidth overflow when multiplying two quantized matrices A and B, which would otherwise require additional processing of the resulting matrix.

At this point, both the quantized base parameters and the low-rank error reconstructor reside in the same low-bitwidth domain, allowing them to be directly merged through

addition. The merged parameters may experience bitwidth overflow. To handle overflow, we adopt the simplest clipping operation, truncating the overflowed portion to the maximum representable bitwidth. The formula is as follows:

$$W_{\text{merged}} = \text{clip}(W_q + (AB)_q, \text{min}, \text{max}), \tag{7}$$

where min and max denote the minimum and maximum representable values of the target bitwidth, respectively.

Traditional methods using full-precision LoRA for error correction on quantized LLMs, which leads to a lossless merging since dequantization and requantization are required at this stage. Compared to this lossless state, our QER retains near-identical accuracy before and after merging, resulting in lossless merging.

D. Error-aware Training

After merging with the quantized low-rank error reconstructor (QER) in low-bitwidth format, the errors of the quantized LLM have been transformed. As there is no need for dequantization and requantization, the original error from precision mismatch in weight merging and requantization loss will no longer occur. Instead, they are now replaced by the following two types of error. First, inherent precision degradation occurs during AB parameter quantization. Second, potential bitwidth overflow during parameter merging necessitates clip operations, leading to truncation-induced errors. To address these errors, we develop an error-aware training on QER that incorporates these errors into the learning process. This implements optimization of both AB quantization errors and overflow clipping errors. Hence, it enables the learning process to adaptively compensate for errors during updates of the AB parameters.

Compared to the original errors, the reconstructed errors offer the advantage during optimization, since both AB quantization errors and overflow clipping errors inherently stem from the AB parameters. Originally, AB parameters should be optimized in the common QAT to mitigate quantization errors on the LLM's full-precision base weights. Within our error-aware training, we simultaneously reduce the precision loss from initial quantization on the base weights, correct the error from quantization on the AB weights, and prevent overflow risks during the merging operation. Finally, we are able to achieve three error reconstructions through the same parameter optimization process.

Moreover, our error-aware training framework on QER maintains the efficiency of standard low-rank training processes. Our optimization mechanism operates without requiring additional computational modules, seamlessly achieving multi-objective error correction through the existing training infrastructure while preserving native computational efficiency. Specifically, the training data employed in our error reconstruction strictly matches the PTQ method used as our initial quantization method. Following established PTQ practices, we utilize minimal calibration data randomly sampled from generic datasets. For example, we follow PTQ approaches like QLLM that employ merely 128 samples from the Wikitext2 dataset. This constrained calibration dataset suffices for our training process, effectively eliminating the need for large-scale training data. Based on the Wikitext2 corpus, we optimize the low-rank AB parameters using the cross-entropy (CE) loss to minimize errors and ensure that the quantized LLM learns the correct labels.

In addition, we use a teacher-student framework for knowledge distillation. The teacher model is the full-precision LLM model. The student is the quantized LLM model. The knowledge distillation loss (KD loss) is calculated as the KL-divergence between the logits of the teacher and student models. This approach transfers knowledge from the high-precision teacher model to the quantized student model, improving performance. The final combined loss function is:

$$\mathcal{L}_{\text{total}} = \lambda \mathcal{L}_{\text{KD}} + \mu \mathcal{L}_{\text{CE}}. \tag{8}$$

Our proposed framework is able to bridge the gap between quantization and low-rank adaptation, enabling more effective low-precision quantization of LLM while preserving their accuracy and efficiency.

IV. Experiments

A. Experimental Setup

Datasets. To evaluate the performance of our QER, we evaluate the perplexity, a key indicator of a model's generative performance that correlates significantly with zero-shot outcomes, on WikiText2 and C4. Additionally, we report zero-shot accuracy on various benchmarks, including PIQA, ARC, HellaSwag and WinoGrande.

Models and Initial PTQ methods. We apply our QER to quantize the LLaMA-1 [2] and LLaMA-2 [3] models. We apply QLLM as an initial PTQ method to quantize the LLaMA-1 and LLaMA-2.

Compared methods. We conduct comparative experiments with diverse quantization methods.

- We compare the initial PTQ method QLLM [15]. For fair comparisons, we use official implementations of the method to establish our initial quantized model. This initial baseline solely undergoes PTQ quantization without any finetuning. We then proceed to apply our QER framework. Direct comparisons demonstrate performance gains after QER application.
- We compare our QER with other recent PTQ quantization methods, such as OmniQuant [10], SmoothQuant (SQ) [8], and Outlier Suppression+ (OS+) [11]. Results for these methods are sourced from QLLM [15].
- We compare our QER with weight-activation QAT methods, including LLM-QAT [12] and QLLM with error correction. In all experimental results tables, "QLLM (PTQ)" is the result of PTQ without any training, and "QLLM" is the result of error correction after PTQ.

Quantization settings. Following [10], [12], we quantize all weights and intermediate activations, with the exception of the Softmax output probability, which is maintained at full precision. We focus on 4-bit weight-activation quantization.

TABLE I: Performance comparisons of different methods for 4-bit weight-activation quantization on LLaMA-1-7B.

Metric	Dataset	W16A16	W4A4							
			SQ	LLM-QAT	LLM-QAT+SQ	OS+	OmniQuant	QLLM (PTQ)	QLLM	QER (Ours)
PPL↓	WikiText2	5.68	52.85	-	-	40.32	11.26	14.17	9.65	**7.65**
	C4	7.08	104.35	-	-	-	14.51	16.70	12.29	**11.24**
	Avg.	6.38	78.60	-	-	-	12.89	15.43	10.97	**9.45**
Acc. (%)↑	PIQA	77.37	49.80	51.50	55.90	62.73	66.15	63.87	68.77	**70.46**
	ARC-e	52.48	30.40	27.90	35.50	39.98	45.20	41.37	45.20	**47.10**
	ARC-c	41.38	25.80	23.90	26.40	30.29	31.14	28.58	31.14	**35.15**
	HellaSwag	72.99	27.40	31.10	47.80	44.39	56.44	53.21	57.43	**61.48**
	Winogrande	66.93	48.00	51.90	50.60	52.96	53.43	52.17	56.67	**60.06**
	Avg.	62.23	36.28	37.26	43.24	46.07	50.47	48.68	51.84	**54.85**

TABLE II: Performance comparisons of different methods for 3-bit weight-activation quantization on LLaMA-1-7B.

#Bits	Method	PPL ↓			Accuracy (%) ↑					
		WikiText2	C4	Avg.	PIQA	ARC-e	ARC-c	HellaSwag	Winogrande	Avg.
W16A16	-	5.68	7.08	6.38	77.37	52.48	41.38	72.99	66.93	62.23
W3A3	QLLM	22966.50	15535.44	19250.97	49.08	26.01	27.39	25.69	49.57	35.55
	Ours	**34.79**	**59.26**	**47.02**	**54.03**	**32.87**	**26.71**	**32.60**	**51.46**	**39.53**

Implementation details. Recent excellent PTQ methods construct their calibration sets using 128 randomly sampled sequences from WikiText2, each with a sequence length of 2048. Therefore, during our QER fine-tuning process, we also use only the same 128 data samples, identical to the calibration set.

It is worth noting that our QER fine-tuning process executes for **only a single epoch**, meaning it traverses the 128 data points just once, without cycling through the data.

The loss coefficients are set to $\lambda = 0.2$ and $\mu = 1$. We use the KL divergence temperature of T = 5. The LoRA component operates at rank 4. AdamW with a linear learning rate decay scheduler is used following QLLM. The learning rate is set to 5×10^{-4}. All training experiments are conducted on a single NVIDIA A800 80G GPU. We use the Language Model Evaluation Harness toolbox [21] for evaluation.

B. Main Results

1) Results on LLaMa-1-7B: Table I presents a comprehensive evaluation of 4-bit weight-activation quantization methods on the LLaMA-1-7B model, comparing perplexity (PPL, lower is better) on WikiText2 and C4 datasets alongside zero-shot accuracy (Acc., higher is better) across five benchmarks (PIQA, ARC-e, ARC-c, HellaSwag, Winogrande), with lower PPL and higher accuracy indicating superior performance. Our proposed Quantized Low-rank Error Reconstructor (QER) achieves the lowest average PPL of 9.45 (7.65 on WikiText2 and 11.24 on C4), significantly outperforming all baselines: it reduces PPL by 13.8% compared to QLLM (10.97), which applies error correction after post-training quantization (PTQ), and by 26.7% compared to the initial PTQ method without error correction QLLM (PTQ) (15.43). In accuracy, QER attains an average of 54.85% across all tasks, exceeding QLLM (51.84%) by 3.01 percentage points and QLLM (PTQ) (48.68%) by 6.17 percentage points, while also surpassing specialized methods like OmniQuant (50.47%) and LLM-QAT+SQ (43.24%) by 4.38 and 11.61 percentage points, respectively. Notably, QER excels in complex reasoning tasks, achieving 61.48% on HellaSwag (vs. QLLM's 57.43%) and 60.06% on Winogrande (vs. QLLM's 56.67%), highlighting its robustness in contextual understanding. These results underscore QER's superiority in addressing the precision mismatch issue inherent in low-rank adaptation for quantized LLMs. By enabling lossless low-bitwidth merging and error-aware training, QER effectively eliminates dequantization and requantization errors, thereby validating our research hypothesis and establishing state-of-the-art performance for 4-bit quantization with minimal performance degradation.

Table II presents a rigorous evaluation of 3-bit weight and activation quantization on the LLaMA-1-7B model, demonstrating the exceptional effectiveness of the proposed Quantized Low-rank Error Reconstructor (QER) in mitigating the severe degradation inherent to ultra-low bitwidth regimes. Under 3-bit quantization, where precision reduction causes exponentially greater distortion than 4-bit settings, QER achieves dramatic improvements across both perplexity (PPL) and accuracy metrics. For the QLLM baseline, which collapses catastrophically with average PPL soaring to 19250.97 and accuracy plummeting to 35.55%, our method reduces average PPL by 99.8% (19250.97 $\rightarrow$ 47.02) and boosts accuracy by 11.2% (35.55% $\rightarrow$ 39.53%). These findings establish QER's ability to recover extreme information loss in 3-bit quantization.

2) Results on LLaMa-2-7B: Table III demonstrates the superior performance of our QER for 4-bit weight-activation quantization on the LLaMA-2-7B model, significantly out-

TABLE III: Performance comparisons of different methods for 4-bit weight-activation quantization on LLaMA-2-7B.

Metric	Dataset	W16A16	W4A4					
			SQ	OS+	OmniQuant	QLLM (PTQ)	QLLM	QER (Ours)
PPL↓	WikiText2	5.47	101.77	-	14.61	17.39	11.75	**8.07**
	C4	6.97	93.21	-	18.39	22.24	13.26	**11.79**
	Avg.	6.22	97.49	-	16.50	19.81	12.51	**9.93**
Acc. (%)↑	PIQA	76.82	60.17	63.11	65.94	62.46	67.68	**68.77**
	ARC-e	53.62	35.23	39.10	43.94	39.90	44.40	**45.75**
	ARC-c	40.53	27.13	28.84	30.80	32.34	30.89	**34.04**
	HellaSwag	72.87	37.08	47.31	53.53	50.44	58.45	**62.11**
	Winogrande	67.25	49.57	51.30	55.09	53.20	56.59	**58.01**
	Avg.	62.22	41.84	45.93	49.86	47.67	51.60	**53.74**

TABLE IV: Performance comparisons of different methods for 3-bit weight-activation quantization on LLaMA-2-7B.

#Bits	Method	PPL ↓			Accuracy (%) ↑					
		WikiText2	C4	Avg.	PIQA	ARC-e	ARC-c	HellaSwag	Winogrande	Avg.
W16A16	-	5.47	6.97	6.22	76.82	53.62	40.53	72.87	67.25	62.22
W3A3	QLLM	33542.11	27000.57	30271.34	49.73	25.76	**27.99**	25.57	49.80	35.77
	Ours	**54.04**	**98.64**	**76.34**	**52.88**	**30.39**	24.91	**28.99**	**51.70**	**37.77**

TABLE V: Performance comparisons of different training strategies.

#Bits	Method	PPL ↓			Accuracy (%) ↑					
		WikiText2	C4	Avg.	PIQA	ARC-e	ARC-c	HellaSwag	Winogrande	Avg.
W16A16	-	5.68	7.08	6.38	77.37	52.48	41.38	72.99	66.93	62.23
W4A4	QAT w/o QER (10 epoch)	11.40	14.20	12.80	68.23	44.57	33.36	57.73	52.72	51.32
	QAT w/o QER (1 epoch)	12.95	15.74	14.34	54.77	42.26	34.64	54.77	51.78	47.64
	QER (Ours)	**7.65**	**11.24**	**9.45**	**70.46**	**47.10**	**35.15**	**61.48**	**60.06**	**54.85**

performing all baseline methods. QER achieves the lowest perplexity across both WikiText2 (8.07 vs. QLLM's 11.75) and C4 (11.79 vs. QLLM's 13.26), reducing average PPL by 20.6% compared to the best alternative (9.93 vs. 12.51). Crucially, QER achieves state-of-the-art accuracy across all reasoning benchmarks, with an average accuracy of 53.74% that exceeds QLLM by 2.14 percentage points and OmniQuant by 3.88 points.

Table IV demonstrates the role of QER in recovering severely degraded quantizations on 3-bit LLaMA-2-7B. These results prove the unique capacity of our QER to reconstruct models from quantization failure states.

C. Ablation Studies

We conduct ablation studies on the LLaMA-1-7B model to assess the contributions of QER's core components.

1) Different Training Strategies: A vanilla QAT (Quantization-Aware Training) approach can attempt to address the issue of new errors introduced by requantization after error correction by integrating the final quantization step into the training process. However, this vanilla method still suffers from lossy merging due to the absence of our quantized low-rank error reconstructor. We term this vanilla QAT baseline "QAT w/o QER". To rigorously validate QER's superiority, we conducted a critical ablation study comparing our QER against "QAT w/o QER". Both methods used only 128 calibration samples. When trained for just one epoch, "QAT w/o QER" exhibits minimal error-correction capability. Notably, even after 10 epochs, it still performs much worse than our QER trained for just one epoch. The results in Table V validate the structural advancements of our QER, whose error reconstructor and low-bitwidth alignment avoid precision-mismatch penalties inherent in vanilla QAT.

TABLE VI: Verification of lossless merging.

#Bits	Method	PPL ↓		
		WikiText2	C4	Avg.
W4A4	QLLM before Merging	9.35	11.93	10.64
	QLLM after Merging	9.65	12.29	10.97
	Merging Loss ↓	0.3	0.36	0.33
	Ours before Merging	7.70	11.21	9.46
	Ours after Merging	7.65	11.24	9.45
	Merging Loss ↓	**-0.05**	**0.03**	**-0.01**

2) Verification of Lossless Merging: Table VI provides direct empirical evidence to verify the lossless merging capability of our QER, with results explicitly reflecting the performance differences between QER and the baseline QLLM during the weight merging process. Merging loss here is measured by changes in PPL, where lower values indicate

TABLE VII: Inference efficiency.

#Bits	W4A4	
Method	QLLM	Ours
Tokens/s	7,781	7,881

better model quality. Specifically, positive merging loss values mean the model's PPL increases after merging, signifying performance degradation, while negative values imply PPL decreases, indicating improved performance after merging. The results show QLLM suffers significant degradation during merging (0.33 average PPL increase), demonstrating its fundamentally lossy process. In contrast, QER achieves near lossless merging. This validates the core innovation of our QER that it avoids the destructive dequantization-requantization cycle that plagues existing methods.

D. Inference Efficiency

To evaluate the inference efficiency of the model quantized by our QER framework, we compared its inference speed with that of the initial PTQ methods in Table VII. The quantized models from our QER and its initial PTQ method perform equivalent operations during inference, as they adopt the same low-bitwidth arithmetic and memory footprint. For example, QER and QLLM achieve nearly identical latency and throughput. This parity confirms that QER's performance gains stem from its enhanced error correction capability, preserving deployment efficiency while improving accuracy.

V. Conclusion

Our proposed Quantized Low-rank Error Reconstructor (QER) addresses the bitwidth mismatch issue when combining low-rank adaptation with quantization-aware training for LLM models, which introduces additional errors during merging and requantization. The QER enables lossless merging in low-bitwidth format by aligning the bitwidth of low-rank parameters with the quantized base model and adopts error-aware training to optimize quantization and overflow errors. Experimental results on LLaMA models validate the superiority of QER, demonstrating its effectiveness for high-accuracy low-precision LLM deployment.

References

[1] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[2] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," *ArXiv*, vol. abs/2302.13971, 2023.

[3] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[4] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph convolutional module for temporal action localization in videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6209–6223, 2021.

[5] M. Yang, E. C. Ngai, X. Hu, B. Hu, J. Liu, E. Gelenbe, and V. C. Leung, "Digital phenotyping and feature extraction on smartphone data for depression detection," *Proceedings of the IEEE*, 2025.

[6] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, "Gptq: Accurate post-training quantization for generative pre-trained transformers," *arXiv preprint arXiv:2210.17323*, 2022.

[7] J. Lin, J. Tang, H. Tang, S. Yang, X. Dang, and S. Han, "Awq: Activation-aware weight quantization for llm compression and acceleration," *arXiv preprint arXiv:2306.00978*, 2023.

[8] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, "Smoothquant: Accurate and efficient post-training quantization for large language models," in *International Conference on Machine Learning*. PMLR, 2023, pp. 38 087–38 099.

[9] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, "Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 30 318–30 332, 2022.

[10] W. Shao, M. Chen, Z. Zhang, P. Xu, L. Zhao, Z. Li, K. Zhang, P. Gao, Y. Qiao, and P. Luo, "Omniquant: Omnidirectionally calibrated quantization for large language models," *arXiv preprint arXiv:2308.13137*, 2023.

[11] X. Wei, Y. Zhang, Y. Li, X. Zhang, R. Gong, J. Guo, and X. Liu, "Outlier suppression+: Accurate quantization of large language models by equivalent and optimal shifting and scaling," *arXiv preprint arXiv:2304.09145*, 2023.

[12] Z. Liu, B. Oguz, C. Zhao, E. Chang, P. Stock, Y. Mehdad, Y. Shi, R. Krishnamoorthi, and V. Chandra, "Llm-qat: Data-free quantization aware training for large language models," *arXiv preprint arXiv:2305.17888*, 2023.

[13] C. Zeng, S. Liu, Y. Xie, H. Liu, X. Wang, M. Wei, S. Yang, F. Chen, and X. Mei, "Abq-llm: Arbitrary-bit quantized inference acceleration for large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 21, 2025, pp. 22 299–22 307.

[14] W. Zhao, Y. Shi, X. Lyu, W. Sui, S. Li, and Y. Li, "ASER: activation smoothing and error reconstruction for large language model quantization," in *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, T. Walsh, J. Shah, and Z. Kolter, Eds. AAAI Press, 2025, pp. 22 822–22 830.

[15] J. Liu, R. Gong, X. Wei, Z. Dong, J. Cai, and B. Zhuang, "QLLM: accurate and efficient low-bitwidth quantization for large language models," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[16] Y. Xu, L. Xie, X. Gu, X. Chen, H. Chang, H. Zhang, Z. Chen, X. Zhang, and Q. Tian, "Qa-lora: Quantization-aware low-rank adaptation of large language models," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[17] M. Chen, W. Shao, P. Xu, J. Wang, P. Gao, K. Zhang, and P. Luo, "Efficientqat: Efficient quantization-aware training for large language models," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Association for Computational Linguistics, 2025, pp. 10 081–10 100.

[18] Z. Liu, C. Zhao, I. Fedorov, B. Soran, D. Choudhary, R. Krishnamoorthi, V. Chandra, Y. Tian, and T. Blankevoort, "Spinquant: LLM quantization with learned rotations," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[19] H. Lin, H. Xu, Y. Wu, J. Cui, Y. Zhang, L. Mou, L. Song, Z. Sun, and Y. Wei, "Duquant: Distributing outliers via dual transformation makes stronger quantized llms," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds., 2024.

[20] Z. Liu, B. Oguz, C. Zhao, E. Chang, P. Stock, Y. Mehdad, Y. Shi, R. Krishnamoorthi, and V. Chandra, "LLM-QAT: data-free quantization aware training for large language models," in *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, L. Ku, A. Martins, and V. Srikumar, Eds. Association for Computational Linguistics, 2024, pp. 467–484.

[21] L. Gao, J. Tow, S. Biderman, S. Black, A. DiPofi, C. Foster, L. Golding, J. Hsu, K. McDonell, N. Muennighoff, J. Phang, L. Reynolds, E. Tang, A. Thite, B. Wang, K. Wang, and A. Zou, "A framework for few-shot language model evaluation," Sep. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5371628