RILQ: Rank-Insensitive LoRA-Based Quantization Error Compensation for Boosting 2-Bit Large Language Model Accuracy

Geonho Lee[1*], Janghwan Lee[1*], Sukjin Hong[2*], Minsoo Kim[1], Euijai Ahn[2], Du-Seong Chang[3], Jungwook Choi[1†]

[1]Hanyang University
[2]KT Corporation
[3]Sogang University
{thisisho, hwanii0288}@hanyang.ac.kr, sukjin.hong@kt.com, minsoo2333@hanyang.ac.kr, euijai.ahn@kt.com, dschang@sogang.ac.kr, choij@hanyang.ac.kr

Abstract

Low-rank adaptation (LoRA) has become the dominant method for parameter-efficient LLM fine-tuning, with LoRA-based quantization error compensation (LQEC) emerging as a powerful tool for recovering accuracy in compressed LLMs. However, LQEC has underperformed in sub-4-bit scenarios, with no prior investigation into understanding this limitation. We propose *RILQ* (Rank-Insensitive LoRA-based Quantization Error Compensation) to understand fundamental limitation and boost 2-bit LLM accuracy. Based on rank analysis revealing model-wise activation discrepancy loss's rank-insensitive nature, *RILQ* employs this loss to adjust adapters cooperatively across layers, enabling robust error compensation with low-rank adapters. Evaluations on LLaMA-2 and LLaMA-3 demonstrate *RILQ*'s consistent improvements in 2-bit quantized inference across various state-of-the-art quantizers and enhanced accuracy in task-specific fine-tuning. *RILQ* maintains computational efficiency comparable to existing LoRA methods, enabling adapter-merged weight-quantized LLM inference with significantly enhanced accuracy, making it a promising approach for boosting 2-bit LLM performance.

Appendix — https://arxiv.org/pdf/2412.01129

Introduction

Large language models (LLMs) like GPT-4 (OpenAI 2023) and LLaMA-3 (Meta 2024) have revolutionized various domains, demonstrating human-level performance in complex tasks such as question answering (Kamalloo et al. 2023), code auto-completion (Rozière et al. 2024), and summarization (Zhang et al. 2024b). However, adapting these models to specialized domains requires efficient fine-tuning techniques (Wei et al. 2022a; Wang et al. 2023). *Low-rank adaptation* (LoRA) (Hu et al. 2022) has emerged as a leading solution, efficiently reparameterizing weight matrices with low-rank adapters to incorporate task-specific information. By fine-tuning only a small, adaptable extension to the frozen base model, LoRA significantly reduces the memory footprint while effectively specializing LLMs. This cost-effective fine-tuning approach has expanded to enable autonomous module composition (Huang et al. 2023), multiple-task adaptation (Sheng et al. 2023), and long-context inference (Chen et al. 2024). LoRA has emerged as a powerful tool for quantization error compensation in large language models (LLMs), addressing the challenges of reduced inference and fine-tuning costs (Dettmers et al. 2023). While weight quantization techniques (Frantar et al. 2023; Lee et al. 2023; Yao et al. 2022; Lin et al. 2023) mitigate LLMs' memory footprint, they introduce errors due to reduced-precision representation. Therefore, LoRA has been adopted to help compensate quantization error. Fig. 1 illustrates two LoRA-based quantization error compensation (LQEC) approaches: (1) Direct error correction, where adapters offset quantization errors in weights. For instance, ZeroQuant-v2 (Yao et al. 2023) employs a low-rank adapter per linear module, which can be merged into quantized weights (Xu et al. 2024; Liu et al. 2024) for efficient inference (Fig. 1(a)). (2) Task-specific fine-tuning, exemplified by LoftQ (Li et al. 2024), which combines LoRA

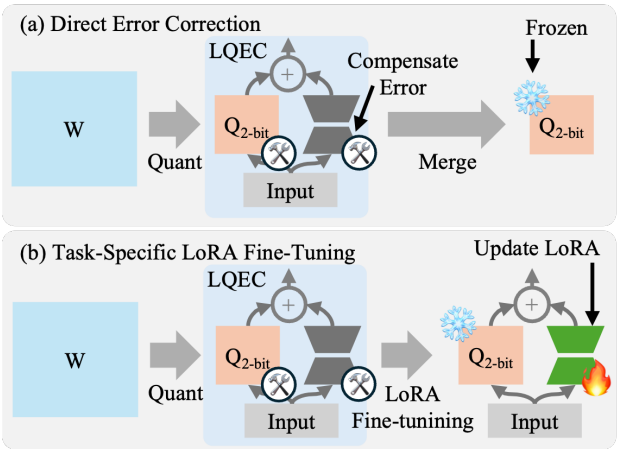

Figure 1: LoRA-based quantization error compensation (LQEC): (a) direct error correction, (b) initialization for task-specific fine-tuning.

*These authors contributed equally.
†Corresponding author.

with the base model quantized and frozen for memory-efficient fine-tuning, using LQEC-tuned adapters as initialization (Fig. 1(b)). Notably, LQEC integrates into existing structures without requiring additional adapters, offering a promising method to address LLM's memory bottleneck while preserving accuracy.

Despite advances in LQEC, achieving high compression rates, such as 2-bit quantization, without compromising model accuracy remains challenging. Aggressive quantization, particularly at sub-4-bit precision, often leads to significant accuracy degradation. Various initiatives have attempted to bridge this accuracy gap by minimizing quantization-induced discrepancies. Weight-level approaches like LQ-LoRA (Guo et al. 2024), LoftQ, LQER (Zhang et al. 2024a), RA-LoRA (Kim et al. 2024), and RoLoRA (Huang et al. 2024b) address discrepancies at each weight quantization by factorizing errors into low-rank adapters via singular value decomposition (SVD). ApiQ (Liao and Monz 2024) compensates for quantization error via gradient updates based on losses at each linear module's output, while QLLM (Liu et al. 2024) and ReALLM (Leconte et al. 2024) use losses at the output of Transformer decoder layers. While these approaches have partially mitigated quantization errors through various ad-hoc techniques, they still suffer significant accuracy losses at 2-bit levels. Moreover, there remains a lack of fundamental understanding as to why LQEC struggles to compensate for aggressive bit-precision quantization, highlighting the need for in-depth analysis.

In this work, we explore the challenges of LQEC under 2-bit weight quantization, highlighting that *lower bit-precision necessitates higher ranks for effective error compensation*, which contradicts LoRA's low-rank premise. We introduce rank sensitivity analysis to assess the rank requirements for error compensation and find that *rank sensitivity decreases as discrepancy scope increases*. Leveraging this insight, we propose *RILQ* (Rank-Insensitive LoRA-based Quantization Error Compensation), which employs a model-wise discrepancy loss at the output of the last Transformer layer. This approach enables cooperative adjustment of both rank-redundant and rank-critical linear modules during LoRA tuning, facilitating flexible signal propagation and quantization error compensation at the model output. Our method delivers robust LQEC even with small ranks (e.g., 16-rank), recovering accuracy while preserving the efficiency of existing LQEC techniques. Evaluations on LLaMA-2 and LLaMA-3 demonstrate consistent improvements in 2-bit quantized inference across state-of-the-art quantizers (OmniQuant (Shao et al. 2024), QuIP (Chee et al. 2023; Tseng et al. 2024), QuaRot (Ashkboos et al. 2024)) and enhanced accuracy in task-specific fine-tuning without additional inference cost, enabling efficient adapter-merged weight-quantized LLM inference of QA-LoRA (Xu et al. 2024) with significant accuracy boosts. These results suggest that our method effectively repositions LQEC as a promising accuracy enhancer for 2-bit LLM inference.

Related Work

LLM Weight Quantization. Weight quantization is a promising technique to reduce the memory footprint of LLMs by lowering the bit-precision of weight values (Frantar et al. 2023; Lee et al. 2023; Yao et al. 2022; Lin et al. 2023; Wei et al. 2022b, 2023). Activation-aware weight quantization methods (Lin et al. 2023; Lee et al. 2024; Kim et al. 2023b; Guo et al. 2023; Heo et al. 2023) have successfully reduced weight precision to 4-bit or lower, but these approaches incur mixed-precision overhead and struggle with poor accuracy at 2-bit precision. Parallel efforts have focused on developing advanced quantizers for 2-bit LLMs (Yao et al. 2022; Shao et al. 2024; Chee et al. 2023; Tseng et al. 2024; Guan et al. 2024; Egiazarian et al. 2024). For instance, OmniQuant introduced learnable quantization parameters to adjust weight ranges and transformations, QuIP employed grouped vector quantization with a shared codebook for non-uniform weight group representation, and QuaRot rotated the weight matrix to redistribute outliers. While these non-uniform quantization schemes achieve robust 3$\sim$4-bit LLM inference accuracy, they still face challenges in closing the accuracy gap for 2-bit LLM inference.

LoRA for Fine-tuning and Compensation. Low-rank adaptation (LoRA) has emerged as the leading parameter-efficient fine-tuning technique to enhance the capabilities of foundational large language models (LLMs) (Hu et al. 2022; Huang et al. 2023; Xia, Qin, and Hazan 2024; Hu et al. 2023; Ding et al. 2022; Han et al. 2024; Chen et al. 2023a; Sheng et al. 2023). LoraHub (Huang et al. 2023) aggregates LoRA modules trained on different tasks to autonomously compose compatible modules, while SLoRA (Sheng et al. 2023) facilitates multiple-LoRA blocks for various tasks, and LongLoRA (Chen et al. 2024) enables context extension for long-context LLM inference. LoRA has also been adapted for *error compensation*, with ZeroQuant-v2 employing a low-rank adapter to compensate for weight quantization errors and similar approaches used for pruning error compensation (Li, Tang, and Zhang 2024; Zhang et al. 2023; Chen et al. 2023b). Despite these advances, LoRA-based error compensation methods still face challenges in achieving high compression rates, such as 2-bit quantization, without compromising model accuracy.

Quantization Error Compensation. Quantization error compensation (QEC) has been extensively explored in the context of quantization-aware training (QAT), which supports aggressive sub-4bit quantization while preserving task-specific fine-tuned accuracy (Kim et al. 2023a; Liu et al. 2023). For example, TSLD (Kim et al. 2023a) uses full-parameter tuning combined with token-wise scaled knowledge distillation to enable robust 2-bit LLM inference, though it requires significant memory for full-parameter adjustments, similar to pre-training. To reduce the memory demands of QAT, LoRA-based parameter-efficient QEC have been developed (Dettmers et al. 2023; Xu et al. 2024; Kim et al. 2024; Guo et al. 2024; Li et al. 2024; Liao and Monz 2024; Chai et al. 2023). Notably, QLoRA (Dettmers et al. 2023) applies LoRA on top of frozen quantized weights to minimize memory usage, while QA-LoRA integrates adapters into quantized weights to enhance LLM inference

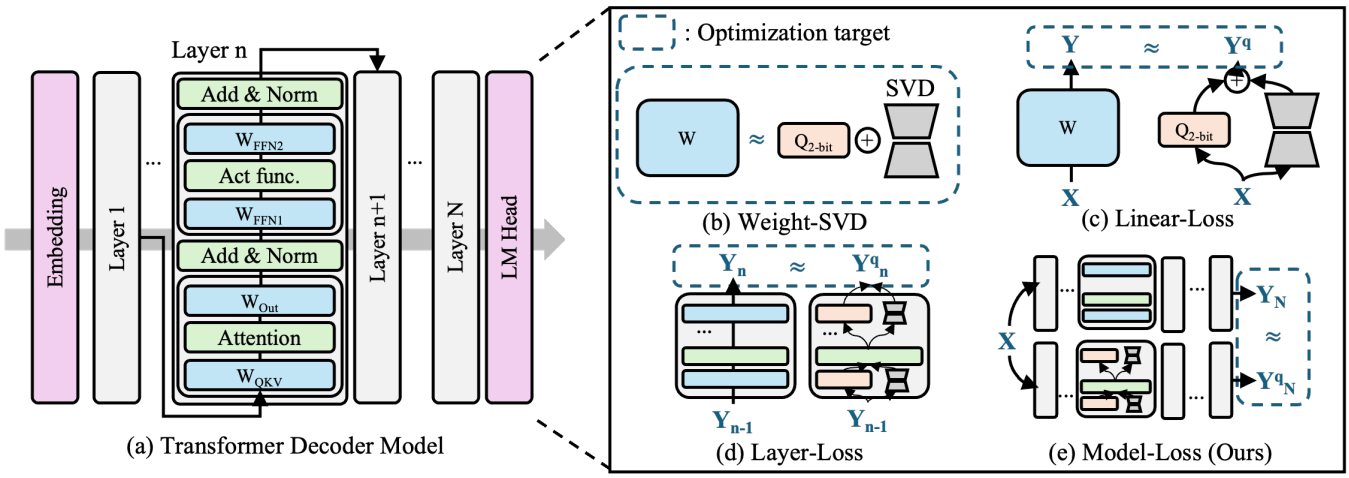

Figure 2: (a) Structure of the Transformer decoder model. (b-e) Four optimization approaches for fine-tuning LoRA for quantization error compensation.

efficiency. Despite these advancements, significant accuracy losses still occur with extremely low-bit quantization. Recent quantization-aware LoRA methods, such as LoftQ as well as (Guo et al. 2024; Zhang et al. 2024a; Kim et al. 2024; Huang et al. 2024b) tackle these challenges by using singular value decomposition (SVD) to factorize quantization errors into low-rank adapters. (Liao and Monz 2024; Liu et al. 2024; Leconte et al. 2024) address quantization errors through gradient updates based on losses at each linear module or Transformer decoder layer's output. Despite these efforts, maintaining accuracy at 2-bit precision remains a significant challenge with no understanding about why.

Background and Challenges

LoRA-based Quantization Error Compensation

LLMs typically comprise multiple Transformer decoder layers, an Embedding layer for token-to-activation conversion, and an LM-Head for converting Transformer outputs into vocabulary logits (Fig.2(a)). Each Transformer layer contains several linear modules with weight parameters ($W_{QKV}, W_{Out}, W_{FFN1}, W_{FFN2}$) for matrix multiplication with input (X) to compute output activation (Y). Since the number of parameters exceeds billions, weight quantization, a prominent model compression technique, represents a pre-trained weight matrix $W \in \mathbb{R}^{d_1 \times d_2}$ using a limited bit width b. The quantized weight Q_b is formulated by Eq. 1:

$$\begin{aligned} Q_b &= s \cdot \text{clamp}(\left\lfloor \frac{W}{s} \right\rceil - z, 0, 2^N - 1) + z \\ s &= \frac{\gamma \max(W) - \beta \min(W)}{2^b - 1}, z = \left\lfloor \frac{\beta \min(W)}{s} \right\rceil, \end{aligned} \tag{1}$$

where $\lfloor \cdot \rceil$ indicates the rounding function. The choice of γ and β varies depending on the quantizers; they can be a constant ($\gamma = 1, \beta = 1$) for the round-to-nearest quantization (RTN) or represent the learnable clipping strengths for the upper and lower bounds of quantized weights as in OmniQuant.

LoRA introduces learnable low-rank adapter modules, $L_1 \in \mathbb{R}^{d_1 \times r}$ and $L_2 \in \mathbb{R}^{d_2 \times r}$ ($r \ll \max(d_1, d_2)$), with frozen pre-trained W, formulating the forward operation for a linear module in the Transformer layer as $Y = X(W + L_1 {L_2}^T)$. For LoRA-based QEC, rank-r adapters are updated to compensate for the impact of weight quantization. For example, LoftQ updates the adapters to minimize the weight discrepancy (Weight-SVD, Fig. 2(b)):

$$\arg\min_{L_1, L_2} \|W - (Q_b + L_1 L_2^\top)\|_F, \tag{2}$$

where $Q_b = \text{Quant}(W - L_1 L_2^\top)$, and $L_1 L_2^\top$ is iteratively updated via SVD. However, weight discrepancy optimization does not consider the combined impact of W and X to the matrix multiplication output Y. Thus, (Liao and Monz 2024) proposed the discrepancy optimization on the output of linear modules (Linear-Loss, Fig. 2(c)):

$$\arg\min_{L_1, L_2} \|Y - Y^q\|_F, \tag{3}$$

where $Y = WX$ and $Y^q = (Q_b + L_1 L_2^\top)X$. QLLM further extended the optimization scope to a Transformer layer by updating L_1 and L_2 parameters within it via gradient from the loss between Y_n and Y_n^q defined as :

$$\begin{aligned} Y_n &= G(Y_{n-1}, \{W_{n,l}\}_{l=1}^L) \\ Y_n^q &= G(Y_{n-1}^q, \{Q_{b,n,l}, L_{1,n,l}, L_{2,n,l}\}_{l=1}^L), \end{aligned} \tag{4}$$

where n is a layer index and G represents a group of L linear modules within a Transformer layer which are sequentially processed according to the Transformer structure. As described in Fig. 2(d), this layer-wise discrepancy loss (Layer-Loss) asserts the alignment of quantized output activation to the FP16 output at each Transformer layer.

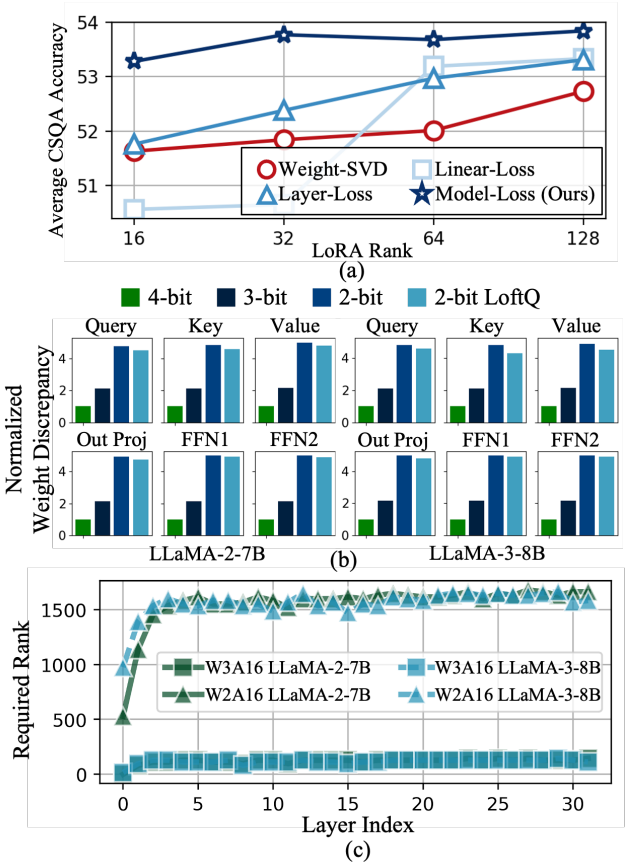

Figure 3: (a) Average CSQA accuracy across optimization granularity and the rank of LoRA (LLaMA-2-7B). (b) Normalized weight discrepancy ($\|W - Q\|_F$) across models (LLaMA-2-7B and LLaMA-3-8B) and every linear module, normalized to 1 for 4-bit quantization discrepancy. (c) Minimum rank required for each quantization bit-precision to closely achieve the weight discrepancy of 4-bit quantization.

LQEC's Challenge for 2-bit LLM

Despite efforts to compensate for quantization errors at various scopes, LQEC has shown limited success with aggressive sub-4-bit quantization. Fig. 3(a) demonstrates that existing discrepancy minimization approaches (Weight-SVD, Linear-Loss, Layer-Loss) suffer growing accuracy loss as rank decreases when LLaMA-2-7B is quantized to 2-bit, contradicting LoRA's fundamental assumption of low-rank fine-tuning updates. To understand these limitations for 2-bit LLM inference, we investigate quantization error characteristics. Fig. 3(b) reveals that weight discrepancy between full-precision and quantized weights increases significantly as bit-precision decreases from 4-bit to 2-bit, with a notable *jump* at 2-bit. This pattern is consistent across weight types and models (LLaMA-2 and LLaMA-3), persisting even with LoftQ adapter initialization. These findings suggest that quantization errors are substantially exacerbated at 2-bit precision, challenging the effectiveness of current LQEC methods.

Method	Relative Error (LM-Head output)			
	r=16	r=32	r=64	r=128
Linear-Loss	4.54	4.49	4.21	3.85
Layer-Loss	4.14	4.06	3.82	3.77
Model-Loss (Ours)	2.85	2.59	2.46	2.59

Table 1: Relative error of the LM-head output activation compared to the baseline inference across error compensation strategies (weights are quantized using OmniQuant).

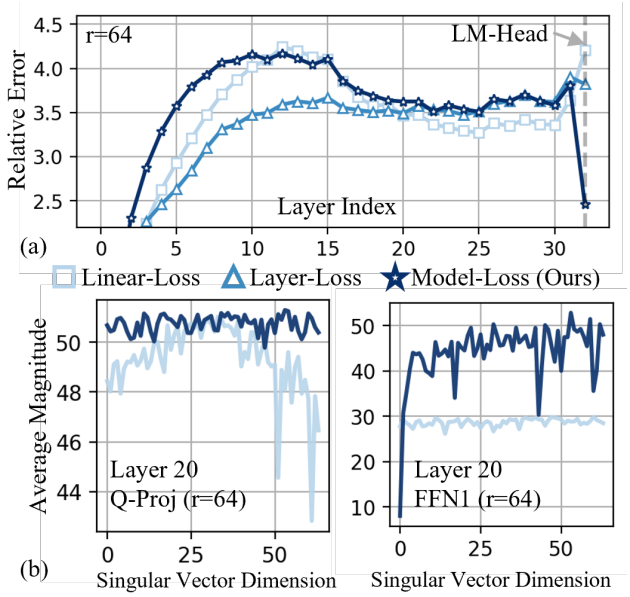

Figure 4: (a) Relative error of intermediate activations and head output compared to baseline inference. (b) Comparison of average magnitudes of left singular vectors between linear and model level optimization.

We further investigate why previous LQEC methods like Weight-SVD of LoftQ (which assume low-rank quantization errors) failed to mitigate the significant discrepancies at 2-bit quantization. While this assumption holds for 3-4-bit quantization, as supported by LQER and SqueezeLLM (Kim et al. 2023b), its validity for 2-bit quantization remained unexplored. Fig. 3(c) illustrates the minimum rank required to suppress weight discrepancy across different bit-precisions. Notably, 3-bit quantization needs only a small adapter rank, but 2-bit quantization demands a much higher rank. This finding suggests that *2-bit quantization errors are inherently high-rank*, challenging the effectiveness of typical SVD-based low-rank adaptation techniques used in existing LQEC methods.

Methodology

In this section, we first propose the rank sensitivity analysis to reveal the impact of the discrepancy scope on LQEC performance. Based on this intriguing finding, we propose a simple yet effective loss, model-level discrepancy loss (Model-Loss), as a new objective for LQEC that overcomes

the rank sensitivity of 2-bit quantization errors.

Rank Sensitivity Analysis

The main limitation of existing LQEC methods is their indiscriminate use of low-rank adapters for QEC without considering quantization error characteristics. To address this, we introduce a new metric called *rank sensitivity*, which measures the relative error ($E = |(Y - Y^q)/Y|$) of logits at the LM-Head output. Lower rank sensitivity (smaller relative error) indicates more accurate inference, as logits directly influence token prediction accuracy. Using this metric, we analyze how the discrepancy scope affects LQEC performance. We extend the discrepancy scope to encompass all Transformer layers, proposing a new discrepancy loss at the output activation of the final (N'th) Transformer layer (Model-Loss, Fig. 2(e)):

$$\underset{L_1,L_2}{\arg\min} \|Y_N - Y_N^q\|_F. \tag{5}$$

Table 1 compares the rank sensitivity of LQEC methods with varying discrepancy scopes and ranks on LLaMA-2-7B. Two key observations emerge: 1) Rank sensitivity decreases as the discrepancy scope expands from a single linear module to the entire model. 2) With Model-Loss, rank sensitivity remains low even at very small ranks (e.g., rank 16). This suggests that Model-Loss mitigates the high-rank requirements typically associated with 2-bit quantization errors.

To elucidate the rank-insensitive nature of Model-Loss, we compare layer-wise relative errors across three scopes of discrepancy loss in LLaMA-2 (Fig. 4(a)). Linear-Loss shows higher relative errors at each Transformer layer compared to Layer-Loss, which is expected given the latter's objective of minimizing layer-wise discrepancy. Notably, Model-Loss exhibits even higher relative errors in intermediate layers, but significantly lower error at the LM-Head. This suggests that internal activation drift may facilitate closer alignment of the final activation with error-free full-precision activation, crucial for accurate token generation insensitive to rank. This insight into enhanced signal propagation flexibility aligns with observations in (Kim et al. 2023a) for full-parameter QAT. However, our Model-Loss incorporates low-rank aspects, necessitating a deeper understanding of parameter-efficient error compensation.

We hypothesize that the rank-insensitivity of Model-Loss stems from its wider discrepancy scope, enabling global adapter adjustment to balance compensation between rank-critical and rank-redundant linear modules during LoRA tuning. This concept builds on RA-LoRA's finding that linear modules have varying rank demands for QEC (e.g., Query-projection (Q-Proj) as low-rank, FFN1 as high-rank (Kim et al. 2024)). Unlike RA-LoRA's complex rank adjustment method, Model-Loss implicitly balance this skewed rank sensitivity during adapter tuning. To support this hypothesis, we compare the average magnitude of each element of LoRA's singular vectors between Linear-Loss and Model-Loss (Fig. 4(b)). Model-Loss significantly increases the overall magnitudes of FFN1's singular vectors compared to Q-Proj's singular vectors. This suggests that Model-Loss enhances the contribution of singular vectors in rank-critical modules (FFN1) while activating previously idle singular vectors in rank-redundant modules (Q-Proj), fostering cooperative quantization error compensation across Transformer layers. More results corresponding to Fig. 4(b) are provided in the Appendix.

Rank-Insensitive LQEC

Building on insights from the rank-insensitive characteristics of Model-Loss, we introduce Rank-Insensitive LoRA-based Quantization error compensation (*RILQ*), a novel method for compensating quantization errors in 2-bit LLMs. *RILQ* significantly improves accuracy by implementing Model-Loss and optimizing LoRA adapters in Transformer layers' linear modules using a Model-Loss. As shown in Eq. 5, *RILQ* uses gradient descent to collectively tune all adapters, minimizing the discrepancy between full-precision and quantized activation outputs ($Y_N - Y_N^q$) of the final layer. This approach effectively addresses inter-weight inconsistencies arising from 2-bit quantization by learning global discrepancy loss from a holistic model perspective. Notably, *RILQ* is particularly advantageous when adapter shapes are constrained to merge with quantized weights for efficient inference (like QA-LoRA), offering a comprehensive solution to LQEC challenges.

To further enhance the language modeling capabilities of LLMs during autoregressive token generation, *RILQ* incorporates a causal language modeling objective with Ground Truth (GT), in the optimization of low-rank adapters (GT-Loss):

$$\underset{\{L_1,L_2\}\in\theta}{\arg\max} \sum_{t=1}^{T} P\left(x_t \mid x_{<t}; \theta\right), \tag{6}$$

where x represents a token and T the sequence length. While this approach has been utilized in previous QAT methods (Kim et al. 2023a), we found it particularly effective in guiding low-rank adapters to improve the model's generation of coherent and contextually appropriate text sequences. This enhancement further aligns 2-bit quantization adjustments with the calibration data. The additional benefits of this method are demonstrated in Table 8 through an ablation study, and the entire procedure for *RILQ* is detailed in the Appendix.

Experiments

Experimental Setup

Tasks and Models. We evaluate our proposed method for common-sense QA tasks (WinoGrande (Sakaguchi et al. 2019), PIQA (Bisk et al. 2019), Hellaswag (Zellers et al. 2019), ARC_challenge (ARC-C) (Clark et al. 2018), ARC_easy (ARC-E) (Clark et al. 2018)) and arithmetic reasoning task (GSM8K (Cobbe et al. 2021)). We focus on two recent open-source pre-trained LLMs, LLaMA-2-7B (Touvron et al. 2023) and LLaMA-3-8B, for evaluating *RILQ*. Additional model scales are in Table 10 in the ablation study.

Quantization Methods. For a comprehensive performance comparison, we employ *RILQ* alongside state-of-the-art weight quantization techniques, including Omni-

Model	Method	Bit-width	*RILQ*	Zero-Shot CSQA Accuracy ↑						Perplexity ↓	
				WG	PIQA	HS	Arc-c	Arc-e	Avg.	WikiText2	C4
LLaMA-2-7B	16-bit Baseline			69.06	78.07	57.14	43.43	76.30	64.80	5.47	6.97
	LoftQ	W2A16	-	51.14	53.97	27.81	21.25	31.14	37.06	1078.24	728.63
			✓	**57.38**	**66.32**	**41.51**	**27.47**	**55.01**	**49.54**	**13.28**	**14.12**
	OmniQuant		-	59.12	70.18	43.14	28.84	58.12	51.88	11.19	12.42
			✓	**62.83**	**72.47**	**46.66**	**31.66**	**63.97**	**55.52**	**9.18**	**10.70**
	QuIP#		-	62.51	71.60	43.84	31.48	62.46	54.38	8.90	11.25
			✓	**66.61**	**75.03**	**51.68**	**37.29**	**69.99**	**60.12**	**6.94**	**8.69**
	QuaRot		-	55.72	62.51	36.42	22.44	49.96	45.41	11.83	19.56
			✓	**62.12**	**72.47**	**47.41**	**30.38**	**64.81**	**55.44**	**7.57**	**10.21**
	OmniQuant	W3A16	-	67.64	77.75	54.97	40.78	**74.49**	63.13	6.07	7.54
			✓	**68.35**	**78.02**	**55.19**	**41.98**	74.20	**63.55**	**6.02**	**7.51**
	QuIP#		-	67.01	76.22	54.45	40.02	75.13	62.57	6.01	7.60
			✓	**67.64**	**76.93**	**55.80**	**40.61**	**75.76**	**63.35**	**5.85**	**7.40**
	QuaRot		-	67.72	76.99	54.12	40.78	**74.79**	62.88	5.91	7.68
			✓	**67.88**	**77.20**	**55.52**	**41.81**	74.62	**63.41**	**5.81**	**7.53**
LLaMA-3-8B	16-bit Baseline			72.61	79.71	60.19	50.43	80.09	68.61	6.14	8.88
	LoftQ	W2A16	-	47.75	53.81	25.91	20.39	26.14	34.80	56168.18	15016.89
			✓	**55.64**	**65.02**	**37.11**	**22.44**	**47.31**	**45.50**	**27.37**	**29.22**
	OmniQuant		-	51.85	59.03	32.86	19.28	36.07	39.82	61.79	52.91
			✓	**58.17**	**69.48**	**42.66**	**27.99**	**56.90**	**51.04**	**17.42**	**20.40**
	QuIP#		-	62.35	67.46	44.66	30.80	57.11	52.48	12.74	16.84
			✓	**66.54**	**74.54**	**52.48**	**38.23**	**71.25**	**60.61**	**9.39**	**12.96**
	QuaRot		-	60.62	65.23	36.77	24.57	57.91	49.02	14.95	27.77
			✓	**68.03**	**73.12**	**48.59**	**33.70**	**68.69**	**58.43**	**10.13**	**15.48**

Table 2: Direct error compensation results. Following the original work, we apply GPTQ (Frantar et al. 2023) on QuaRot. End-to-end fine-tuning is not applied to QuIP#. Results for QuIP# with fine-tuning are presented in Table 9.

Method	*RILQ*	CSQA Tasks↑			GSM8K↑
		PIQA	Arc-c	Arc-e	
16-bit LoRA Fine-Tuning		79.05	47.70	79.25	38.97
OmniQuant	-	77.31	38.31	69.95	28.66
	✓	**77.91**	**40.61**	**71.70**	**30.86**
QuIP#	-	78.07	44.62	74.75	35.56
	✓	**78.45**	**45.99**	**76.01**	**36.32**

Table 3: Task-specific fine-tuning results for CSQA tasks and GSM8K on LLaMA-2-7B.

Inference Bit-width	*RILQ*	Error Compensation			Fine-Tuning
		CSQA↑	Wiki2↓	C4↓	GSM8K↑
W2A16	-	47.42	12.92	9.23	17.89
	✓	**54.51**	**10.65**	**8.14**	**23.73**

Table 4: QA-LoRA implementation of 2-bit LLaMA-2-7B with *RILQ*. Weights are quantized with OmniQuant (Calibration set: WikiText-2), evaluated with accuracy (CSQA, GSM8K) and perplexity (WikiText-2, C4).

Quant, QuIP#, QuaRot, as well as LoftQ (Weight-SVD based LQEC). Each method is sourced from its respective repository. For OmniQuant and QuaRot, we set the group size to 64 (QuIP# employs a codebook). Other implementation details of each quantization method are specified in the Appendix.

***RILQ* Implementation Details**. The *RILQ* implementation includes a calibration process for initializing LoRA adapters on quantized models using the C4 dataset (Raffel et al., 2019). Perplexity is evaluated on both the WikiText-2 (Merity et al., 2016) and C4 datasets. For this purpose, a sequence length 512 is employed, and 256 sentences are randomly sampled from the C4 training dataset. During optimization, the Model-Loss and the GT-Loss are applied, with gradient-based optimization performed using the Adam optimizer. The learning rate is fixed at 1e-4, and the batch size is 8. Additional details, including the overall procedure of *RILQ*, the fine-tuning settings, and an analysis of memory costs, are provided in the Appendix.

Experimental Results

Direct Error Correction Results. We evaluate the quantization error compensation capability of *RILQ* on LLaMA-2-7B and LLaMA-3-8B for LoftQ and advanced weight quantization techniques. Table 2 presents CSQA tasks' accuracy and the perplexity of WikiText-2 and C4.

Our key observations are as follows:

- LoftQ (based on Weight-SVD) is inadequate for 2-bit quantization, as it experiences over 19% average accuracy loss, making it unsuitable for deployment. On the other hand, advanced weight quantization methods (OmniQuant, QuIP#, and QuaRot) help recover accuracy.
- *RILQ* significantly enhances the accuracy of weight quantization methods. OmniQuant, QuIP#, and QuaRot experience a 10-19% average accuracy loss, but *RILQ* recovers this by a significant margin.
- *RILQ* is more effective on LLaMA-3-8B than on

Quantization Method	Bit-width	Rank	LQEC	Zero-Shot Accuracy ↑						Perplexity ↓	
				WG	PIQA	HS	Arc-c	Arc-e	Avg.	WikiText2	C4
NormalFloat (Dettmers et al. 2023)	W2A16	16	SVD	51.07	55.33	28.19	19.80	30.85	37.05	1311.39	771.04
			RILQ	**55.01**	**66.59**	**39.59**	**25.60**	**53.32**	**48.02**	**14.92**	**15.34**
		32	SVD	49.96	53.97	27.12	21.33	29.29	36.33	1379.42	872.47
			RILQ	**57.62**	**66.10**	**39.83**	**24.32**	**50.55**	**46.97**	**13.91**	**14.79**
		64	SVD	51.14	53.97	27.81	21.25	31.14	37.06	1078.24	728.63
			RILQ	**57.38**	**66.32**	**41.51**	**27.47**	**55.01**	**49.54**	**13.28**	**14.12**
		128	SVD	49.01	56.31	29.82	22.61	33.67	38.28	437.65	347.13
			RILQ	**57.46**	**69.80**	**43.12**	**28.24**	**56.73**	**51.07**	**11.59**	**12.96**
		256	SVD	49.25	54.41	28.59	20.82	29.92	36.60	471.68	239.91
			RILQ	**60.14**	**70.35**	**44.65**	**29.95**	**59.55**	**52.93**	**10.38**	**11.98**
OmniQuant		16	SVD	59.35	70.13	42.46	27.99	58.21	51.63	11.45	12.71
			RILQ	**61.96**	**73.01**	**46.77**	**31.83**	**64.14**	**55.54**	**9.24**	**10.79**
		32	SVD	58.96	70.35	42.91	28.58	58.42	51.84	11.34	12.50
			RILQ	**62.59**	**73.07**	**47.24**	**31.91**	**64.10**	**55.78**	**9.29**	**10.83**
		64	SVD	58.48	70.62	43.48	28.75	58.71	52.01	10.96	12.19
			RILQ	**62.83**	**72.47**	**46.66**	**31.66**	**63.97**	**55.52**	**9.18**	**10.70**
		128	SVD	59.91	70.62	44.11	29.27	59.72	52.73	10.47	11.70
			RILQ	**62.35**	**72.85**	**46.29**	**31.14**	**63.89**	**55.30**	**9.17**	**10.70**
		256	SVD	**61.80**	71.65	45.90	**31.66**	62.16	54.63	9.56	10.99
			RILQ	61.48	**73.01**	**46.31**	**31.66**	**63.59**	**55.21**	**9.17**	**10.72**

Table 5: Ablation study on rank sensitivity comparison between SVD and *RILQ*. Accuracy and PPL are measured right after the quantization error compensation, not after task-specific fine-tuning. (NF2 & SVD = LoftQ) (Model: LLaMA-2-7B).

LQEC	Bit	Rank					σ
		16	32	64	128	256	
SVD	W3A16	7.56	7.55	7.53	7.50	7.45	**0.04**
	W2A16	12.71	12.50	12.19	11.70	10.99	*0.69*
RILQ	W3A16	7.52	7.50	7.50	7.50	7.50	**0.01**
	W2A16	10.79	10.83	10.87	10.70	10.72	**0.07**

Table 6: C4 PPL↓ with different LQEC and bit-width (OmniQuant, LLaMA-2-7B).

Method	Zero-Shot Accuracy ↑			
	PIQA	ARC-C	ARC-E	Avg.
QA-LoRA (Baseline)	73.83	34.56	65.53	57.97
RA-LoRA	74.54	36.18	66.33	59.02
RILQ	**76.39**	**36.86**	**68.98**	**60.74**

Table 7: Comparison of RA-LoRA and *RILQ* under QA-LoRA setting for task-specific fine-tuning for CSQA. RTN is used for 2-bit weight quantization (LLaMA-2-7B, rank=16).

LLaMA-2-7B. Existing LoftQ and weight quantization methods suffer greater accuracy loss on LLaMA-3-8B, consistent with recent observations (Huang et al. 2024a) that LLaMA-3 is more sensitive to quantization. For example, QuIP#, the best-performing quantizer in our experiments, experiences higher accuracy degradation on LLaMA-3. In this case, *RILQ* improves the average accuracy of QuIP# by 8.1%, a significant boost.

- For 3-bit weight quantization, the accuracy improvement by *RILQ* is less noticeable since QuIP# and other weight quantization methods already approach FP16 accuracy, leaving less room for improvement.

Task-Specific Fine-Tuning Results. We further evaluate the task-specific fine-tuning accuracy of *RILQ* on LLaMA-2-7B. LoRA without *RILQ* indicates a default LoRA initialization; one of the adapter pairs is initialized in Gaussian distribution, and the other is zero-initialized. For OmniQuant and QuIP#, weights are first quantized, and then the LoRA or *RILQ* is further fine-tuned with CSQA and GSM8K task-specific datasets. Table 3 shows that *RILQ* consistently improves diverse task-specific fine-tuning performances.

Integrating LoRA into Linear Modules. QA-LoRA reduces overhead by merging adapter parameters into quantized weights via shortening the input activation dimension. However, this shortened input dimension can impair error compensation in aggressive quantization scenarios. We demonstrate that *RILQ*'s rank-insensitive nature synergistically enhances LQEC performance. To validate *RILQ* within the QA-LoRA framework, we evaluate OmniQuant-quantized models before and after fine-tuning. Table 4 shows *RILQ* improves perplexity and CSQA accuracy through error compensation. This enables *RILQ* to match the efficiency of adapter-less quantized LLM inference while significantly boosting accuracy.

Ablation Study

Rank Sensitivity. Table 5 compares *RILQ* with SVD for NormalFloat (NF) (Dettmers et al. 2023) and OmniQuant methods. *RILQ* outperforms SVD at all rank levels in both quantization method, notably achieving better perplexity and accuracy with 16 ranks than SVD with 256 ranks in OmniQuant. This demonstrates *RILQ*'s efficiency, delivering superior performance with smaller adapter sizes. Table 6 presents the C4 perplexity of SVD and *RILQ* under

Scope	Loss		Zero-Shot Accuracy ↑					
	Act	GT	WG	PIQA	HS	Arc-c	Arc-e	Avg.
Linear	✓	-	55.56	65.29	36.58	21.50	46.17	45.02
Layer	✓	-	58.25	68.72	43.35	29.78	56.62	51.34
Model	✓	-	61.17	69.59	43.74	30.29	59.05	52.77
	-	✓	62.04	**71.16**	45.13	29.95	58.12	53.28
	✓	✓	**62.59**	70.89	**45.51**	**32.85**	**61.70**	**54.71**

Table 8: Ablation study on expansion of ApiQ with different discrepancy loss scope: GT, Linear, Layer, Model-Loss (GT & Model-Loss=*RILQ*) (Model: LLaMA-2-7B).

Model	QuIP# FT	*RILQ*	Avg. CSQA↑	PPL↓ Wiki2	C4
LLaMA 3-8B	-	-	52.5	12.7	16.8
	-	✓	60.6	9.4	13.0
	✓	-	59.3	9.4	12.8
	✓	✓	**61.3**	**9.1**	**12.4**

Table 9: Comparison with QuIP#-FT (W2A16).

2∼3-bit quantization across varying ranks. We adjust the rank from 16 to 256 and measure the standard deviation (σ) of the perplexity. With SVD initialization, the perplexity of W3A16 remains stable across different ranks due to the minimal quantization error. However, the reconstruction of the significant 2-bit quantization error is highly sensitive to rank variation, aligning with our observation in Fig. 3 that higher ranks are required for handling 2-bit errors. In contrast, *RILQ* initialization exhibits rank-insensitive results across both bit widths.

To further validate *RILQ*'s rank-insensitive nature, we compare it with standard QA-LoRA (uniform rank for all adapters) and RA-LoRA (rank-adjusted based on sensitivity). As shown in Table 7, *RILQ* outperforms both at low rank (rank=16), suggesting effective internal rank management despite using uniform ranks like standard QA-LoRA.

Impact of Scope for Discrepancy Loss. Table 8 presents an ablation study on LLaMA-2-7B, examining how loss types and optimization granularities affect quantization error compensation. Increasing scope from linear module to model level improves accuracy, highlighting the importance of inter-layer interactions within model. Incorporating GT-Loss with Model-Loss, which provides richer information compared to using either loss individually, further enhances performance, surpassing the effectiveness of GT-Loss alone and serving as an effective optimization guide. This improvement in accuracy aligns with findings of QAT (Kim et al. 2023a). The proposed *RILQ* method, which combines these loss objectives at the model level, achieves the best overall performance, highlighting the benefits of global optimization and diverse loss functions for robust error compensation.

QuIP# end-to-end FT with *RILQ*. We additionally evaluate the cross effects of end-to-end fine-tuning in QuIP# (QuIP#-FT) and *RILQ* on LLaMA-3-8B in Table 9. As shown in Table 9, *RILQ* improves QuIP#-FT CSQA accuracy by 2%, highlighting *RILQ* 's ability to enhance fine-tuning beyond standard end-to-end approaches.

# Params	Bit-width	*RILQ*	Perplexity ↓ WikiText2	C4
7B	W2A16	-	1078.24	728.63
		✓	**13.28**	**14.12**
13B		-	59.95	72.77
		✓	**9.56**	**11.21**
70B		-	12.69	16.36
		✓	**6.42**	**8.38**

Table 10: Error compensation results using *RILQ* across different model sizes in the LLaMA-2 families. The quantization method used in this experiment is LoftQ.

LQEC	# of Samples	Sequence Length	PPL Wiki2	C4	Time
-	-	-	433	474	0m
SVD	-	-	176.17	221.23	31m
RILQ	256	128	11.35	13.20	10m
	256	**512**	**10.04**	**11.79**	**37m**
	256	1024	9.86	11.51	1h 39m
	256	2048	9.61	11.25	3h 52m
	256	4096	9.42	11.07	9h 6m
	64	512	10.91	12.78	26m
	512	512	9.83	11.50	1h
	2048	512	9.34	10.98	3h 7m

Table 11: Perplexity and required time for SVD and RILQ (LLaMA-2-7B, 2-bit RTN, rank=16). The default setting is highlighted in bold.

Experiments on Large Models. To evaluate the scalability of *RILQ*, we conduct experiments across the LLaMA-2 family, scaling from 7B to 70B parameters. As shown in Table 10, *RILQ* consistently enhances the perplexity of the quantized models across all sizes, demonstrating the effectiveness of the proposed quantization error compensation technique, even in larger models.

Convergence Time of *RILQ* Based on Calibset. As shown in Table 11, *RILQ* with default setting (256 samples and sequence length of 512) converges in under 40 minutes, matching SVD in speed while achieving lower PPL. Extending the calibration sequence length or number of samples can reduce PPL to 9.34, but requires more than 3 hours, supporting our default choice as sufficient.

Conclusion

In this work, we propose *RILQ*, a novel LoRA-based quantization error compensation method that effectively addresses the challenges of 2-bit weight quantization in large language models. By explicitly employing LoRA for quantization error compensation and utilizing a model-wise activation discrepancy loss, *RILQ* enables robust quantization-error compensation while maintaining computational efficiency. Experiments on LLaMA-2 and LLaMA-3 demonstrate the superiority of *RILQ* in improving 2-bit quantized LLM inference accuracy and fine-tuning performance.

Acknowledgments

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2020-II201373, Artificial Intelligence Graduate School Program (Hanyang University), and 2022-0-00957, Distributed on-chip memory-processor model PIM (Processor in Memory) semiconductor technology development for edge applications) and National Research Foundation of Korea (NRF) (No. RS-2023-00260527).

References

Ashkboos, S.; Mohtashami, A.; Croci, M. L.; Li, B.; Jaggi, M.; Alistarh, D.; Hoefler, T.; and Hensman, J. 2024. QuaRot: Outlier-Free 4-Bit Inference in Rotated LLMs. *arXiv preprint arXiv:2404.00456*.

Bisk, Y.; Zellers, R.; Bras, R. L.; Gao, J.; and Choi, Y. 2019. PIQA: Reasoning about Physical Commonsense in Natural Language. arXiv:1911.11641.

Chai, Y.; Gkountouras, J.; Ko, G. G.; Brooks, D.; and Wei, G.-Y. 2023. INT2.1: Towards Fine-Tunable Quantized Large Language Models with Error Correction through Low-Rank Adaptation. *arXiv*.

Chee, J.; Cai, Y.; Kuleshov, V.; and Sa, C. D. 2023. QuIP: 2-Bit Quantization of Large Language Models With Guarantees. *arXiv*.

Chen, J.; Zhang, A.; Shi, X.; Li, M.; Smola, A.; and Yang, D. 2023a. Parameter-Efficient Fine-Tuning Design Spaces. *arXiv*.

Chen, T.; Ding, T.; Yadav, B.; Zharkov, I.; and Liang, L. 2023b. LoRAShear: Efficient Large Language Model Structured Pruning and Knowledge Recovery. *arXiv*.

Chen, Y.; Qian, S.; Tang, H.; Lai, X.; Liu, Z.; Han, S.; and Jia, J. 2024. LongLoRA: Efficient Fine-tuning of Long-Context Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *arXiv:1803.05457v1*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.

Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Ding, N.; Qin, Y.; Yang, G.; Wei, F.; Yang, Z.; Su, Y.; Hu, S.; Chen, Y.; Chan, C.-M.; Chen, W.; Yi, J.; Zhao, W.; Wang, X.; Liu, Z.; Zheng, H.-T.; Chen, J.; Liu, Y.; Tang, J.; Li, J.; and Sun, M. 2022. Delta Tuning: A Comprehensive Study of Parameter Efficient Methods for Pre-trained Language Models. *arXiv*.

Egiazarian, V.; Panferov, A.; Kuznedelev, D.; Frantar, E.; Babenko, A.; and Alistarh, D. 2024. Extreme Compression of Large Language Models via Additive Quantization. *arXiv*.

Frantar, E.; Ashkboos, S.; Hoefler, T.; and Alistarh, D. 2023. OPTQ: Accurate Quantization for Generative Pre-trained Transformers. In *The Eleventh International Conference on Learning Representations*.

Guan, Z.; Huang, H.; Su, Y.; Huang, H.; Wong, N.; and Yu, H. 2024. APTQ: Attention-aware Post-Training Mixed-Precision Quantization for Large Language Models. *arXiv*.

Guo, C.; Tang, J.; Hu, W.; Leng, J.; Zhang, C.; Yang, F.; Liu, Y.; Guo, M.; and Zhu, Y. 2023. OliVe: Accelerating Large Language Models via Hardware-friendly Outlier-Victim Pair Quantization. *arXiv*.

Guo, H.; Greengard, P.; Xing, E.; and Kim, Y. 2024. LQ-LoRA: Low-rank plus Quantized Matrix Decomposition for Efficient Language Model Finetuning. In *The Twelfth International Conference on Learning Representations*.

Han, Z.; Gao, C.; Liu, J.; Zhang, J.; and Zhang, S. Q. 2024. Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey. *arXiv*.

Heo, J. H.; Kim, J.; Kwon, B.; Kim, B.; Kwon, S. J.; and Lee, D. 2023. Rethinking Channel Dimensions to Isolate Outliers for Low-bit Weight Quantization of Large Language Models. *arXiv*.

Hu, E. J.; yelong shen; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Hu, Z.; Lan, Y.; Wang, L.; Xu, W.; Lim, E.-P.; Lee, R. K.-W.; Bing, L.; and Poria, S. 2023. LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models. *arXiv*.

Huang, C.; Liu, Q.; Lin, B. Y.; Pang, T.; Du, C.; and Lin, M. 2023. LoraHub: Efficient Cross-Task Generalization via Dynamic LoRA Composition. *arXiv*.

Huang, W.; Ma, X.; Qin, H.; Zheng, X.; Lv, C.; Chen, H.; Luo, J.; Qi, X.; Liu, X.; and Magno, M. 2024a. How Good Are Low-bit Quantized LLaMA3 Models? An Empirical Study. arXiv:2404.14047.

Huang, X.; Liu, Z.; Liu, S.-Y.; and Cheng, K.-T. 2024b. RoLoRA: Fine-tuning Rotated Outlier-free LLMs for Effective Weight-Activation Quantization. arXiv:2407.08044.

Kamalloo, E.; Dziri, N.; Clarke, C.; and Rafiei, D. 2023. Evaluating Open-Domain Question Answering in the Era of Large Language Models. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5591–5606. Toronto, Canada: Association for Computational Linguistics.

Kim, M.; Lee, S.; Lee, J.; Hong, S.; Chang, D.-S.; Sung, W.; and Choi, J. 2023a. Token-Scaled Logit Distillation for Ternary Weight Generative Language Models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Kim, M.; Lee, S.; Sung, W.; and Choi, J. 2024. RA-LoRA: Rank-Adaptive Parameter-Efficient Fine-Tuning for Accurate 2-bit Quantized Large Language Models. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics ACL 2024*, 15773–15786. Bangkok, Thailand and virtual meeting: Association for Computational Linguistics.

Kim, S.; Hooper, C.; Gholami, A.; Dong, Z.; Li, X.; Shen, S.; Mahoney, M. W.; and Keutzer, K. 2023b. SqueezeLLM: Dense-and-Sparse Quantization. *arXiv*.

Leconte, L.; Bedin, L.; Nguyen, V. M.; and Moulines, E. 2024. ReALLM: A general framework for LLM compression and fine-tuning. arXiv:2405.13155.

Lee, C.; Jin, J.; Kim, T.; Kim, H.; and Park, E. 2024. OWQ: Outlier-Aware Weight Quantization for Efficient Fine-Tuning and Inference of Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(12): 13355–13364.

Lee, J. H.; Kim, J.; Kwon, S. J.; and Lee, D. 2023. FlexRound: Learnable Rounding based on Element-wise Division for Post-Training Quantization. In Krause, A.; Brunskill, E.; Cho, K.; En-

gelhardt, B.; Sabato, S.; and Scarlett, J., eds., *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, 18913–18939. PMLR.

Li, G.; Tang, Y.; and Zhang, W. 2024. LoRAP: Transformer Sub-Layers Deserve Differentiated Structured Compression for Large Language Models. *arXiv*.

Li, Y.; Yu, Y.; Liang, C.; Karampatziakis, N.; He, P.; Chen, W.; and Zhao, T. 2024. LoftQ: LoRA-Fine-Tuning-aware Quantization for Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Liao, B.; and Monz, C. 2024. ApiQ: Finetuning of 2-Bit Quantized Large Language Model. *arXiv*.

Lin, J.; Tang, J.; Tang, H.; Yang, S.; Dang, X.; and Han, S. 2023. AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration. *arXiv*.

Liu, J.; Gong, R.; Wei, X.; Dong, Z.; Cai, J.; and Zhuang, B. 2024. QLLM: Accurate and Efficient Low-Bitwidth Quantization for Large Language Models. arXiv:2310.08041.

Liu, Z.; Oguz, B.; Zhao, C.; Chang, E.; Stock, P.; Mehdad, Y.; Shi, Y.; Krishnamoorthi, R.; and Chandra, V. 2023. LLM-QAT: Data-Free Quantization Aware Training for Large Language Models. *arXiv*.

Meta. 2024. Llama 3 Model Card.

OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

Rozière, B.; Gehring, J.; Gloeckle, F.; Sootla, S.; Gat, I.; Tan, X. E.; Adi, Y.; Liu, J.; Sauvestre, R.; Remez, T.; Rapin, J.; Kozhevnikov, A.; Evtimov, I.; Bitton, J.; Bhatt, M.; Ferrer, C. C.; Grattafiori, A.; Xiong, W.; Défossez, A.; Copet, J.; Azhar, F.; Touvron, H.; Martin, L.; Usunier, N.; Scialom, T.; and Synnaeve, G. 2024. Code Llama: Open Foundation Models for Code. arXiv:2308.12950.

Sakaguchi, K.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2019. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. *arXiv preprint arXiv:1907.10641*.

Shao, W.; Chen, M.; Zhang, Z.; Xu, P.; Zhao, L.; Li, Z.; Zhang, K.; Gao, P.; Qiao, Y.; and Luo, P. 2024. OmniQuant: Omnidirectionally Calibrated Quantization for Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Sheng, Y.; Cao, S.; Li, D.; Hooper, C.; Lee, N.; Yang, S.; Chou, C.; Zhu, B.; Zheng, L.; Keutzer, K.; Gonzalez, J. E.; and Stoica, I. 2023. S-LoRA: Serving Thousands of Concurrent LoRA Adapters. *arXiv*.

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

Tseng, A.; Chee, J.; Sun, Q.; Kuleshov, V.; and Sa, C. D. 2024. QuIP#: Even Better LLM Quantization with Hadamard Incoherence and Lattice Codebooks. *arXiv*.

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 13484–13508. Toronto, Canada: Association for Computational Linguistics.

Wei, J.; Bosma, M.; Zhao, V.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2022a. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*.

Wei, X.; Zhang, Y.; Li, Y.; Zhang, X.; Gong, R.; Guo, J.; and Liu, X. 2023. Outlier Suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 1648–1665.

Wei, X.; Zhang, Y.; Zhang, X.; Gong, R.; Zhang, S.; Zhang, Q.; Yu, F.; and Liu, X. 2022b. Outlier Suppression: Pushing the Limit of Low-bit Transformer Language Models. *arXiv*.

Xia, W.; Qin, C.; and Hazan, E. 2024. Chain of LoRA: Efficient Fine-tuning of Language Models via Residual Learning. *arXiv*.

Xu, Y.; Xie, L.; Gu, X.; Chen, X.; Chang, H.; Zhang, H.; Chen, Z.; ZHANG, X.; and Tian, Q. 2024. QA-LoRA: Quantization-Aware Low-Rank Adaptation of Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Yao, Z.; Aminabadi, R. Y.; Zhang, M.; Wu, X.; Li, C.; and He, Y. 2022. ZeroQuant: Efficient and Affordable Post-Training Quantization for Large-Scale Transformers. *arXiv*.

Yao, Z.; Wu, X.; Li, C.; Youn, S.; and He, Y. 2023. ZeroQuant-V2: Exploring Post-training Quantization in LLMs from Comprehensive Study to Low Rank Compensation. *arXiv*.

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? In Korhonen, A.; Traum, D.; and Màrquez, L., eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4791–4800. Florence, Italy: Association for Computational Linguistics.

Zhang, C.; Cheng, J.; Constantinides, G. A.; and Zhao, Y. 2024a. LQER: Low-Rank Quantization Error Reconstruction for LLMs. *arXiv preprint arXiv:2402.02446*.

Zhang, M.; Chen, H.; Shen, C.; Yang, Z.; Ou, L.; Yu, X.; and Zhuang, B. 2023. LoRAPrune: Pruning Meets Low-Rank Parameter-Efficient Fine-Tuning. *arXiv*.

Zhang, T.; Ladhak, F.; Durmus, E.; Liang, P.; McKeown, K.; and Hashimoto, T. B. 2024b. Benchmarking Large Language Models for News Summarization. *Transactions of the Association for Computational Linguistics*, 12: 39–57.