RA-LoRA: Rank-Adaptive Parameter-Efficient Fine-Tuning for Accurate 2-bit Quantized Large Language Models

Minsoo Kim[1], **Sihwa Lee**[1], **Wonyong Sung**[2] and **Jungwook Choi**[1]*
[1]Hanyang University, [2]Seoul National University
Seoul, Republic of Korea
{minsoo2333, macto94, choij}@hanyang.ac.kr
{wysung}@snu.ac.kr

Abstract

Deploying large language models (LLMs) with their extensive parameters and high memory demands challenges computational efficiency, particularly in fine-tuning for specific applications with limited resources. Techniques like Low-Rank Adaptation (LoRA) help by training a smaller, modifiable extension of the base model to reduce memory usage. However, combining quantization with LoRA, especially in low-bit scenarios, can lead to performance losses due to quantization errors. Our innovative Rank-Adaptive LoRA (RA-LoRA) addresses this by dynamically adjusting the adapter's rank using rank-subspace analysis, optimizing performance with fewer parameters. We tested RA-LoRA on state-of-the-art LLMs for 2-bit efficient fine-tuning, showing it can improve model accuracy with minimal trainable parameters, marking a leap forward in quantization-aware fine-tuning methods and highlighting the significance of rank dynamics in optimizing quantized LLMs.

1 Introduction

Large language models (LLMs) such as (Touvron et al., 2023; OpenAI, 2023) demonstrate impressive abilities in tasks like translation and summarization. However, their large size presents deployment challenges due to significant memory demands (Gholami, 2021). A technique known as weight quantization compresses model parameters, reducing memory needs and boosting computational efficiency. Although post-training quantization (PTQ) can be conveniently employed for compression of large pre-trained models, it often compromises inference accuracy, particularly in low-bit environments (Dettmers et al., 2022; Frantar et al., 2023; Lin et al., 2023; Kim et al., 2024; Shao et al., 2024; Lee et al., 2023).

The field of natural language processing is increasingly adopting the strategy of fine-tuning large-scale, pre-trained LLMs for specific downstream applications (Wei et al., 2022; Wang et al., 2022). While effective in enhancing model performance, fine-tuning traditionally updates the entire pre-trained model, a method becoming unsuitable with the growing size of LLMs. Hence, the push for parameter-efficient fine-tuning methods is clear, offering the advantage of training a smaller, adaptable extension to the base model and substantially lowering the memory footprint for adaptation. Techniques like low-rank adaptation (LoRA) (Hu et al., 2022), which efficiently reparameterizes weight matrices, lead this initiative by providing a feasible technique to fine-tuning of LLMs in various applications.

Combining quantization with low-rank adaptation techniques, such as QLoRA (Dettmers et al., 2023), lessen the memory footprint for fine-tuning LLMs by melding parameter-efficient fine-tuning (PEFT) with model compression, especially PTQ. Yet, this approach encounters notable hurdles with aggressive low-bit quantization scenarios like 2-bits, where quantization errors frequently cause significant accuracy drops. Despite efforts by methods like LoftQ and LQ-LoRA to overcome these challenges through strategic initialization, issues persist at lower bit levels, emphasizing the need for more sophisticated solutions that preserve both memory efficiency and model accuracy during the fine-tuning of quantized LLMs.

Our investigation reveals that quantization error in LLMs inherently opposes low-rank adaptation. This resistance is especially pronounced under aggressive low-bit quantization, attributed to the high-rank subspace of the quantization error. Importantly, we discovered that vectors with small singular values play a pivotal role in the fine-tuning process for correcting quantization errors, hindering effective, parameter-efficient error compensation via low-rank adaptation. Moreover, our study indicates that the characteristics of quantiza-

* Corresponding Author

Findings of the Association for Computational Linguistics: ACL 2024, pages 15773–15786

tion errors in rank subspace vary depending on their location within the model, specifically in weights of feedforward networks and output projections, and differ across layer numbers.

Based on this understanding, we introduce a novel, parameter-efficient, quantization-aware fine-tuning method known as Rank-Adaptive LoRA (RA-LoRA). This method dynamically adjusts the adapter's rank based on *rank-subspace analysis* to effectively counter quantization errors with fewer parameters during fine-tuning. Our evaluation of RA-LoRA on advanced LLMs like DeBERTa-V3 and LLaMA-2 for 2-bit parameter-efficient fine-tuning demonstrates its capacity to significantly enhance model accuracy with a lower parameter count, highlighting RA-LoRA's superiority over existing quantization-aware fine-tuning approaches.

2 Related Work

2.1 Weight Quantization for LLMs

Weight quantization has been proposed as to alleviate the significant memory constraints associated with generation-based inference tasks in LLMs, with a particular focus on 4-bit PTQ (Frantar et al., 2023; Lin et al., 2023; Kim et al., 2024; Shao et al., 2024; Dettmers et al., 2023) that maintains performance without the need for additional training. However, moving to sub-4-bit quantization introduces challenges related to accuracy degradation due to higher compression rates. To mitigate this issue, quantization-aware training (QAT) for LLMs can be implemented (Liu et al., 2023; Kim et al., 2023). This method reduces the adverse effects of low-bit quantization while necessitating re-training. Yet, as LLMs continue to grow in size, the memory requirements for training pose a barrier to the application of QAT. In response, we propose a parameter-efficient fine-tuning approach capable of effectively mitigating the quantization error associated with the 2-bit quantization of LLMs.

2.2 Low-Rank Adaptation of LLMs

In the rapidly evolving landscape of LLMs, various parameter-efficient fine-tuning (PEFT) strategies have been proposed to mitigate the soaring costs associated with fine-tuning ever-larger models. Among these, the low-rank adaptation (LoRA) (Hu et al., 2022) approach has emerged as a paradigm-shifting direction, allowing for fine-tuning with significantly fewer trainable parameters by introducing adapter matrices A and B while freezing the pre-trained weight W. The foundational assumption of LoRA is that updates applied to pre-trained LLMs during fine-tuning for downstream learning exhibit a low-rank structure. (Hu et al., 2022) demonstrated that a handful of top singular vectors spanning the subspace learned through fine-tuning effectively determine the major direction of the entire adapter matrix, thereby justifying the feasibility of low-rank adaptation. Furthering this exploration, (Aghajanyan et al., 2021) delved into the intrinsic dimensionality of pre-trained language models, revealing that natural language tasks could be learned with few dimensions.

2.3 Quantization-aware LoRA

Building upon the efficient learning potential of LoRA, methods that combine it with quantization such as QLoRA (Dettmers et al., 2023) has been proposed to curtail the costs associated with loading frozen pre-trained models into GPU memory by utilizing quantized LLMs. This approach makes LLM fine-tuning more cost-accessible by utilizing mild 4-bit weight quantization, which largely preserves the original generalization ability of LLMs. (Frantar et al., 2023; Kim et al., 2024; Shao et al., 2024). To elevate memory efficiency further, the concept of sub-4-bit quantization-aware LoRA raises intriguing challenges.

Recent works like LoftQ and LQ-LoRA (Li et al., 2024; Guo et al., 2024) suggest that in the sub-4bit regime, the fixup zero initialization (Zhang et al., 2019) as a default setting of QLoRA introduces a performance weight discrepancy at the starting point of fine-tuning. Consequently, they propose a method of jointly optimizing the quantization of weights and adapter initialization to create an initial point that minimizes the discrepancy, offering a more accurate pathway for fine-tuning. QA-LoRA (Xu et al., 2024) advances toward a completely quantized inference approach by adjusting the dimensions of LoRA parameters and incorporating these with quantization zero-point parameters, thus enabling the direct use of fully quantized weights during the inference stage.

However, prior works have not sufficiently the challenge of downstream learning with extremely low-bit quantized pre-trained models when restricted to a *low-rank subspace*. As evidence of this oversight, all quantization-aware LoRA methods (Li et al., 2024; Guo et al., 2024) applying a uniform rank of 64 across all sublayers, a strategy that

continues to result in significant accuracy degradation in the context of 2-bit quantization. This research aims to investigate the impact of the adapter subspace within extremely low-bit quantization-aware LoRA scenarios and identifies that the optimal rank required to compensate for quantization errors varies significantly among the transformer sublayers. Based on these observations, we will propose a rank adaptation methodology, RA-LoRA aimed at notably improving the accuracy degradation observed in recent quantization-aware LoRA approaches.

3 Backgrounds

3.1 Weight Quantization

Weight quantization discretizes pre-trained weight matrix, $W \in \mathbb{R}^{d_1 \times d_2}$ into a limited number of bits reducing the memory footprint and enabling optimized hardware utilization. Following prior works (Lin et al., 2023; Dettmers et al., 2022), we define $W_Q = Q_N(W)$ with min-max based uniform quantization as shown in Eq. 1 ($\lfloor \cdot \rceil$ represents rounding function).

$$W_Q = Q_N(W) = s\lfloor \frac{W-z}{s} \rceil + z. \quad (1)$$

The scale factor $s = \frac{\max(W)-\min(W)}{2^N-1}$ is determined by minimum or maximum values in each quantization group, where N is the bit-width and the zero-point is $z = \min(W)$. In this study, we focus on the significant accuracy degradation observed in 2-bit quantization, a challenge that persists in the recent quantization methods[1].

3.2 Low-Rank Adapter Initialization

QLoRA (Dettmers et al., 2023) reformulate the linear transformation as $Y = XW_Q + XAB^\top$, where W_Q represents the quantized pre-trained matrix and $A \in \mathbb{R}^{d_1 \times r}, B \in \mathbb{R}^{r \times d_2}$ represent LoRA adapters. Recent approaches, such as LoftQ and LQ-LoRA (Li et al., 2024; Guo et al., 2024) implement a heuristic LoRA adapters initialization method. This involves alternating step between the quantization of pre-trained weights and the application of singular value decomposition (SVD) to minimizing the following objective:

$$\min_{W_Q, A, B} \|W - W_Q - AB^\top\|_F. \quad (2)$$

Specifically, the process is iterative, with two main computations at each t-step[2]. Initially, LoftQ quantize the difference between the pre-trained weight matrix (W) and the LoRA adapters from prior step, $A_{t-1}B_{t-1}^\top$ (note that $A_0B_0 = 0$).

$$W_{Q,t} = Q_N(W - A_{t-1}B_{t-1}^\top). \quad (3)$$

Subsequently, LoftQ perform truncated SVD on the residual quantization error ($W_{qerr,t} = W - W_{Q,t}$) with d being the hidden dimension of the language model and r being pre-defined rank:

$$W_{qerr,t} = \sum_{i=1}^{d} \sigma_{t,i}\mathbf{u}_{t,i}\mathbf{v}_{t,i}^\top \approx A_tB_t^T, \quad (4)$$

$$A_t = [\sqrt{\sigma_{t,1}}\mathbf{u}_{t,1}, ..., \sqrt{\sigma_{t,r}}\mathbf{u}_{t,r}],$$
$$B_t = [\sqrt{\sigma_{t,1}}\mathbf{v}_{t,1}, ..., \sqrt{\sigma_{t,r}}\mathbf{v}_{t,r}]. \quad (5)$$

Here, $\sigma_{t,1} \geq \sigma_{t,2} \geq ... \geq \sigma_{t,d}$ represents the singular values of $W_{qerr,t}$. $\mathbf{u}_{t,i}$ and $\mathbf{v}_{t,i}$ denote the corresponding left and right singular vectors of each singular values at t step. This iterative initialization process reduces the discrepancy by approximating the quantization error W_{qerr}, yet it fix the dimensionality of the LoRA adapter subspace to be low during the fine-tuning phase. We refer to this LoRA adapter initialization method, based on quantized weights, as quantization-aware LoRA.

4 Challenges

Despite previous efforts in quantization-aware LoRA (Dettmers et al., 2023; Li et al., 2024; Guo et al., 2024) to bridge the accuracy gap, 2-bit quantization still results in significant accuracy degradation. To gain fundamental insights into this challenge, we change various factors of quantization-aware LoRA, such as bit-precision (2∼4 bits), adapter ranks (0∼4096), and the number of alternating steps (1∼5), and explore model weight discrepancy and model accuracy degradation. We first found that the quantization error significantly increases at lower bit-precisions (2-bit), and cannot be adequately compensated by the LoRA adapters, even with previously proposed initialization methods.

Fig. 1(a) investigates the discrepancy between the quantized and the original pre-trained weights

[1]For a fair comparison, we set a group size of 64 in every experiment and analysis, where each 64 elements of each quantization group share the same scale factor (s).

[2]LoftQ employs an arbitrary number of steps for adapter initialization optimization. Consistently, we adopt 5 steps for LoftQ in our experiments.

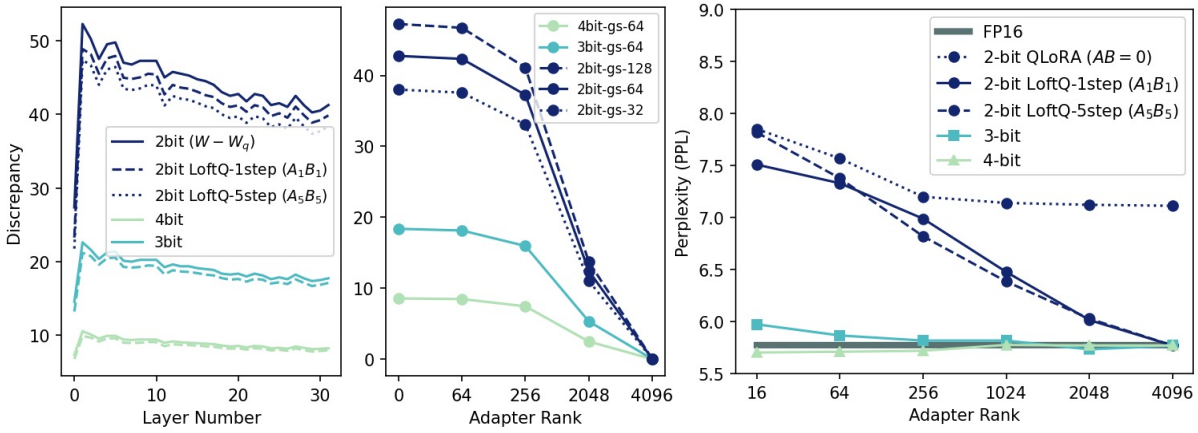


Figure 1: (a) Discrepancy (Frobenius norm) between quantized weight and original pre-trained weight with LoRA adapter with $r = 64$. The dashed line represents LoftQ-1step ($A_1B_{1,r=64}$), and the dotted line represents LoftQ-5step ($A_5B_{5,r=64}$). (b) Weight discrepancy comparison by sweeping adapter rank(r) from 4 to 4096 with LoftQ initialization ($A_5B_{5,r}$). Note that even with fine-grained quantization with group size 32, the trend of requiring a high rank to reduce 2-bit quantized error still exists. (c) LLaMA-2-7B WikiText-2 Quantization-aware LoRA fine-tuning PPL results across QLoRA and LoftQ methods with adapter rank exploration.

in terms of the Frobenius norm across the layers. Notably, the adapter initialization decreases the discrepancy over the steps (from A_1B_1 to A_5B_5), but the reduction is marginal compared to the discrepancy between 2-bit and 3-bit quantization errors. Additionally, Fig. 1(b) examines the effects of raising the adapter rank and reducing the quantization group size, utilizing LoftQ initialization. It highlights a notable discrepancy in all 2-bit quantization scenarios when the rank size is set to 64, a common default rank size for many quantization-aware LoRA methods (Dettmers et al., 2023; Li et al., 2024; Guo et al., 2024).

The observed significant quantization errors in 2-bit quantization directly lead to substantial accuracy degradation during fine-tuning. This is evidenced by the notable gap in perplexity when compared to results from 3-bit and 4-bit fine-tuning, particularly at lower adapter ranks, as demonstrated in Fig. 1(c). Despite adapter initialization (2-bit LoftQ), higher adapter ranks only modestly narrow this gap toward full-precision model performance. This indicates that *2-bit quantization errors are not inherently low-rank*, and attempting to fit these high-rank errors within a low-rank subspace leads to considerable accuracy losses. To recap, without adapter initialization, such as in QLoRA, even with increased ranks, the accuracy gap remains significant. LoftQ's method of initializing quantization errors offers some mitigation, but the accuracy gap fails to reduce substantially when adapter ranks are low, underscoring the need for addressing the high-rank nature of quantization errors to improve accuracy in 2-bit quantized fine-tuning.

5 Analysis

In this section, we explore the effects of adjusting adapter ranks and analyze the learning behaviors of low-rank adapter weights through decomposition into singular values and vectors. This analysis aims to understand how high-rank quantization errors can be compensated within a constrained subspace. These findings motivate us to devise a novel rank adaptation technique that reduces the number of parameters while enhancing fine-tuning accuracy.

5.1 Evolution of Singular Values from Fine-Tuning

How do low-rank adapters learn to compensate for high-rank 2-bit quantization errors? We examine adapter updates during fine-tuning for different ranks (64 and 256), comparing the effects of the initializing strategy of LoftQ ($A_tB_t \approx W_{qerr}$) and QLoRA methods (zero initialization, $A_0B_0 = 0$). This investigation begins by observing singular values $[\sigma_1, ..., \sigma_r]$ of adapters updated across training steps.

Fig. 2 illustrates the development of singular values at ranks 64 (Top) and 256 (Bottom), with both initialized and non-initialized adapters. The analysis sheds light on how singular values evolve during fine-tuning. With LoftQ initialization (Fig. 2(a,b)),

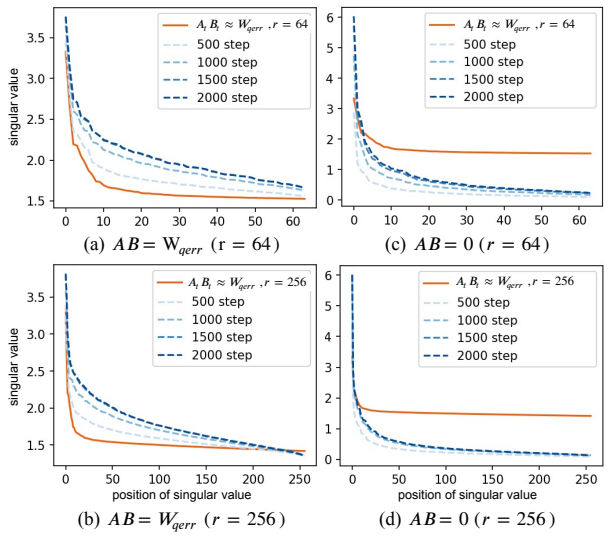


Figure 2: singular values of low-rank adapters across fine-tuning steps. (a),(b) fine-tuning LoftQ-1step initialization. rank 64 and 256 respectively (c)(d) $AB = 0$ rank 64 and 256 respectively. Note that A_1B_1 in (c) and (d) serve as references. (LLaMA-2-7B Up projection layer with GSM8K fine-tuning)

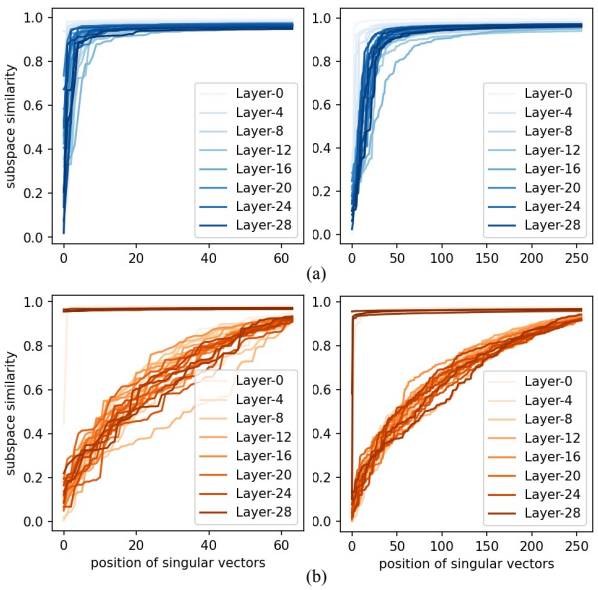


Figure 3: Normalized similarity between quantization error and subspaces spanned by singular vectors of learned adapter per layer. (a) Query projection (b) Up-Proejction, Left: $U_{AB_{r=64}}$ Right: $U_{AB_{r=256}}$ 2-bit quantization-aware GSM8K fine-tuned LLaMA-2-7B model is used for analysis.

an upward trend in singular values is noted across the board for both ranks, implying that adapter learning seeks to augment rank to recover from lost information. In contrast, starting with near-zero singular values in the $AB = 0$ scenario (Fig. 2(c,d)), only a selected few singular values witness substantial growth, notably the top-1 singular value, which exceeds that of $AB = W_{qerr}$. This skewed growth suggests an update concentration, attempting to rectify high-rank quantization errors with limited effectiveness due to the confined rank space. This aligns with Fig. 1(b), where zero initialization leads to suboptimal fine-tuning outcomes.

5.2 Subspace Similarity of Singular Vectors and Quantization Error

Expanding on our analysis of singular values, we delve into the behavior of the corresponding singular vectors, examining the evolution of subspaces they span during fine-tuning across different ranks, especially in terms of compensating for quantization errors. Notably, our observations suggest that *not all sublayers necessitate a high rank* to effectively span a subspace that mitigates the effects of 2-bit quantization errors.

Building on the idea of subspace similarity, which assesses the congruence between subspaces formed by unitary singular vectors from SVD as introduced by (Hu et al., 2022), we evaluate this similarity using the Grassman distance (Hamm and Lee, 2008) between the quantization error matrix (W_{qerr}) and the subspaces delineated by the singular vectors of adapters across various ranks. A subspace similarity approaching 1.0 indicates an increasing alignment of the singular vector subspaces with the quantization error. (Refer to Appendix A.2 for the detailed formulation.)

Fig. 3 illustrates how the similarity between the quantization error and the subspace formed by singular vectors evolves across different ranks. Our analysis reveals considerable diversity in subspace similarity across various sublayers. Notably, the Query layer, depicted in Fig.3 (a), shows that even low-rank singular vectors achieve high similarity with the quantization error across all ranks (64 and 256). Conversely, the Up-projection layer, as seen in Fig.3 (b), indicates a gradual increase in similarity due to contributions from all singular vectors, These findings highlight that the optimal rank for mitigating high-rank quantization errors differs among sublayers, which can be a new direction for exploring parameter-efficienct and accurate fine-tuning. For an expanded analysis incorporating additional models like OPT (Zhang et al., 2022), see Appendix A.2.

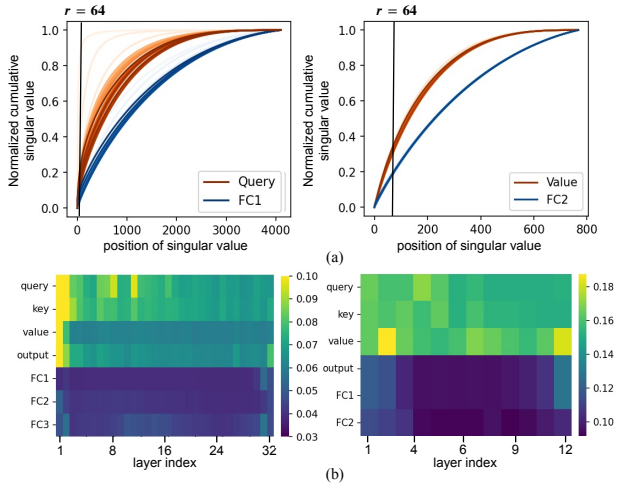


Figure 4: (a) Normalized cumulative singular value of quantization error (W_{qerr}) (b) Heat map of normalized cumulative singular value at the 64th singular value. **Left**:LLaMA-2-7B **right**: DeBERTa-V3-base

Algorithm 1 RA-LoRA Pseudo Algorithm

Input: weights W, target rank r, quantizer $Q_N(\cdot)$, Hyper-parameters α, β, γ with $\alpha > \beta > \gamma$
Output: adapted rank $\boldsymbol{R}$ for each sublayers

```
# Define rank candidates based on r and hyper-parameters
{R_0, R_1, ..., R_5, R_6} ← {r/α, ..., r/β, ..., r/γ}
# Iterate over linear layer weights in the same block
for key in {q, k, v, o, fc_1, fc_2, fc_3} do
    W ← W_key
    W_Q ← Q_N(W)
    # Obtain singular values from the quantization error
    σ ← SVD(W − W_Q)
    # Normalize the singular values
    σ̄ ← σ̄_k = σ_k^2 / ||σ||_2^2
    # Calculate NCSV(c) for target rank(r)
    c_key ← Σ_{k=1}^{r} σ̄_k
end for
# Sort sublayer indices in descending order based on c
keys_sorted ← sort_indices_descending(c)
# Assign rank candidates based on c, inversely proportional to their corresponding NCSV(c)
for i in 0 to 6 do
    R_keys_sorted[i] ← R_i
end for
return R
```

6 Methodology

6.1 Rank Adaptation for Robust Quantization

Motivated by prior insights that not all sublayers require a high rank to span a subspace to mitigate the quantization errors, we adopt a metric called *normalized cumulative singular values* (NCSV, c_{key}) from (Wang et al., 2020) as a measure of the intrinsic rank of quantization errors. As an example, Fig. 4(a) displays NCSV of the Query and Up-Projection layers for two different Transformer models, LLaMA2-7B (Left) and DeBERTa-V3-base (Right). The more the curves are skewed toward the left-top corner, the higher the concentration of the singular values, implying intrinsically low rank. Therefore, given a target rank $r = 64$ (as a default rank of most quantization-aware LoRA techniques), we can assess how effectively a sublayer's quantization error matrix is represented within a given rank's subspace.

Building on the evaluation of NCSV, we aim to compare the extent of ranks needed across different sublayers of the model to compensate for quantization errors. Fig. 4(b) illustrates the heatmap of normalized cumulative singular value across sublayers, revealing the diversity in their rank properties. Notably, it reveals a trend where the quantization errors in self-attention layers are predominantly low-rank, whereas those in MLP layers are generally high-rank. This observation aligns with the subspace similarity discussions in Fig. 3, providing further evidence of the differential rank requirements between these two types of layers. Therefore, we can utilize the normalized cumulative singular values as a guiding metric enabling allocation of effective rank to sublayers for addressing quantization error in fine-tuning [3].

6.2 Rank-Adaptive LoRA

Rank-adaptive LoRA (RA-LoRA) employs a strategy to assign optimal ranks to each sublayer of transformer blocks, utilizing NCSV from the SVD decomposition of the quantization error matrix. By setting a target average rank r, RA-LoRA uses the rth index of these normalized values to determine the rank allocation effectively. The effective rank of each sublayer is adjusted based on the relative size of their singular values, with sublayers having higher values at the target rank receiving lower ranks and vice versa. Algorithm 1 outlines the overall procedure of RA-LoRA[4]. For the LLaMA-2 model, we calculate the normalized cumulative singular values ($\bar{\boldsymbol{\sigma}}$) across the seven sublayers of a transformer block, pinpointing values at indices up to the target rank ($\sum_{k=1}^{r} \bar{\sigma}_k$), as shown in Fig.4(b).

RA-LoRA enhances extremely low-bit quantization-aware LoRA by assigning optimal adapter ranks to each sublayer, effectively mitigat-

[3]Additional ablation studies on the impact of each sublayer ranks on fine-tuning performance can be found in the Appendix.A.3

[4]The implementation of RA-LoRA can effectively utilize the singular values obtained through SVD results from LoftQ. An additional step involves normalizing these values to calculate the NCSV, which incurs a negligible cost.

ing quantization errors. In the following section, we showcase RA-LoRA's effectiveness in 2-bit quantization-aware LoRA, addressing accuracy losses. Additionally, we extend RA-LoRA to the QA-LoRA (Xu et al., 2024) approach for comprehensive 2-bit quantized inference, highlighting gains in accuracy and efficiency.

7 Experiments

We evaluate the effectiveness of our RA-LoRA approach through comparative analysis of task-specific fine-tuning performance on both Natural Language Understanding(NLU) and Natural Language Generation(NLG) tasks. For NLU, we utilize the GLUE (Wang et al., 2019) dataset, focusing on specific tasks (CoLA, RTE, MRPC, STSB) that exhibit significant accuracy drops in 2-bit quantization-aware LoRA (Dettmers et al., 2023; Li et al., 2024). For NLG, we measure perplexity (PPL) using the language modeling benchmark of wikitext (Merity et al., 2016), accuracy with language generation on GSM8K (Cobbe et al., 2021), and zero-shot reasoning accuracy using the commonsense QA tasks (Bisk et al., 2019; Zellers et al., 2019; Clark et al., 2018). The models employed are DeBERTa-v3 (He et al., 2021) and LLaMA-2 (Touvron et al., 2023).

Baseline. The performance of RA-LoRA is evaluated against state-of-the-art quantization-aware LoRA methods. This includes QLoRA (Dettmers et al., 2023) and LoftQ (Li et al., 2024), which finetunes FP16 adapters on 2-bit quantized pre-trained weights and QA-LoRA (Xu et al., 2024), whcih merges adapter weights with quantized weights while ensuring that the entire weight configuration remains within the quantized space, enhancing efficiency during inference.

Trainable Parameter Ratio. For rigorous comparison, we include the trainable parameter ratio alongside performance metrics. This approach allows us to directly compare the efficiency and effectiveness of our rank adaptation method against static rank approaches. In evaluating rank allocation efficiency, we present the percentage of learnable adapter parameters to the total model parameters. Note that our baselines are applying fixed rank 32 in the encoder-only model and 64 in the decoder-only model. (Further detailed training settings are provided in Appendix A.1.)

Method (r)	Trainable (%) ↓	RTE (Acc)↑	CoLA (MCC)↑	MRPC (F1)↑	STSB (F1)↑	Avg. ↑
FP16 LoRA (32)	2.88	82.04	69.20	90.82	91.02	83.27
QLoRA (32)	2.88	60.23	54.05	87.05	87.02	72.09
QLoRA (128)	10.41	60.35	55.15	87.08	87.07	72.41
QLoRA (256)	18.85	61.37	57.81	87.85	87.16	73.55
LoftQ (32)	2.88	62.26	60.57	87.15	87.05	74.26
LoftQ (128)	10.41	64.44	63.81	88.55	87.68	76.12
LoftQ (256)	18.85	71.84	65.92	90.53	88.45	79.19
RA-LoRA (32)	1.98	63.48	61.02	88.54	88.02	75.27
RA-LoRA (128)	7.29	63.85	64.74	90.21	87.66	76.62
RA-LoRA (256)	14.41	**74.05**	**66.89**	**90.96**	**88.68**	**80.15**

Table 1: Results of 2-bit quantization-aware LoRA fine-tuning (QLoRA, LoftQ and RA-LoRA) on the development sets of RTE, CoLA, STSB, and MRPC with DeBERTa-v3-base. Report median number over five random seeds. (r) is target rank.

7.1 Evaluation on Encoder Model

Table 1 details the performance improvements in fine-tuning the DeBERTaV3-base (He et al., 2021) pre-trained model on GLUE tasks, specifically targeting accuracy recovery from high-rank 2-bit quantization errors. We explore different ranks for each methodology, beginning with 32 as utilized by LoftQ and extending to 128 and 256, to assess enhancements in performance.

The performance comparison reveals the improved parameter-efficiency of RA-LoRA thanks to rank adaptation. For QLoRA, as analyzed in Sec 5.1, we observe no significant improvement in fine-tuning performance with increased ranks, and its performance remains inferior to LoftQ. In contrast, LoftQ shows performance improvements as the rank increases, consistent with our observations from Fig. 1(b). Applying rank adaptation (RA-LoRA) yields superior performance, achieving average of 76.62 at a lower ratio of 7.29% compared to LoftQ's 10.41% with an average 76.12. Even at higher ranks, RA-LoRA exhibits significant accuracy improvements moving closer to FP performance reaching an average of 80.15. These results confirm the significance of rank adaptation in 2-bit fine-tuning in the encoder-only model, highlighting the efficacy of RA-LoRA in narrowing the performance gap in 2-bit quantization.

7.2 Evaluation on Decoder Model

Table 2 showcases the fine-tuning results for LLaMA-2 on Wikitext and GSM8K tasks, highlighting RA-LoRA's strengths. Across all tasks and models, RA-LoRA consistently outperforms QLoRA and LoftQ in terms of perplexity (PPL) and task accuracy, demonstrating its significant advantage. Notably, RA-LoRA's rank adaptation enables

Method (r)	Trainable (%) ↓	LLaMA-2-7B Wikitext (PPL) ↓	LLaMA-2-7B GSM8K (Acc) ↑	LLaMA-2-13B Wikitext (PPL) ↓	LLaMA-2-13B GSM8K (Acc) ↑
FP16 LoRA (64)	2.37, 1.92	5.77	34.80	5.44	43.10
QLoRA (64)	2.37, 1.92	7.59	18.53	6.55	20.85
LoftQ (64)	2.37, 1.92	7.45	21.76	6.46	30.17
LoftQ (256)	9.49, 7.70	7.14	25.85	6.40	34.69
RA-LoRA (64)	2.14, 1.73	7.33	23.42	6.46	34.69
RA-LoRA (256)	8.37, 6.98	**7.07**	**28.96**	**6.29**	**35.90**

Table 2: Results of 2-bit quantization-aware LoRA fine-tuning (QLoRA, LoftQ and RA-LoRA) on Wikitext and GSM8K. (PPL) results (the lower, the better) from Wikitext development set and (Acc)uracy results from GSM8K test set with LLaMA-2-7b/13b. (r) is target rank.

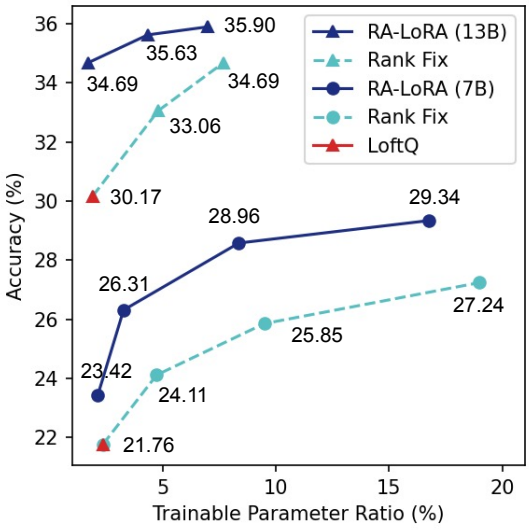


Figure 5: Comparison of 2-bit GSM8K fine-tuning accuracy with LLaMA-2-7B and LLaMA-2-13B across trainable parameter ratios, showcasing RA-LoRA superior accuracy over Rank Fix strategy (Li et al., 2024) (Guo et al., 2024) at all trainable parameter ratios.

it to achieve up to a 3% performance increase on the LLaMA-2-7b GSM8K task while using fewer parameters than LoftQ's trainable parameter count of 2.37. This underscores RA-LoRA's potential in leveraging effective rank based on the sublayer's property on quantization errors, allowing for superior performance with a lower parameter budget.

Figure 5 showcases RA-LoRA's performance enhancements across various training budgets, highlighting the limitations of a uniform rank application across all sublayers (Rank Fix), which often yields inferior performance gains. This limited performance gains arise from not accounting for each sublayer's unique rank characteristics due to quantization errors. Conversely, RA-LoRA's strategy of allocating ranks based on these characteristics

Method (r)	Trainable Params (%) ↓	GSM8K Accuracy (%) ↑	Fine-Tuning Speed (iter/sec) ↑	Fine-Tuning Memory (MiB) ↓
Full QAT	100	22.52	1.50	67146
QA-LoRA (64)	1.31	21.30	3.13	22302
QA-LoRA (256)	5.23	21.38	2.57	28528
RA-LoRA (64)	1.07	**21.38**	3.13	22176
RA-LoRA (256)	5.07	**23.20**	2.60	28052

Table 3: QA-LoRA fine-tuning results on GSM8K with and without RA-LoRA with LLaMA-2-7b. Average iterations per second and maximum allocated memory throughout the entire fine-tuning process. (r) is target rank.

ensures singnificant improvements in performance as the trainable parameter ratio increases. Notably, RA-LoRA achieves a significant performance leap in LLaMA-2-7B, with an accuracy of 26.31 at a 3.28% parameter ratio, surpassing LoftQ's accuracy of 25.85 at a 9.49% parameter ratio. Furthermore, in the LLaMA-2-13B model, RA-LoRA dramatically reduces the required parameter ratio by four times, achieving the same accuracy of 34.69 with just 1.73% of learnable parameters, compared to LoftQ's 7.70%, thereby demonstrating the efficacy of targeted rank allocation.

7.3 Pushing Further for Fully Quantized 2-bit Quantization-Aware Training (QAT)

We extend the application of RA-LoRA to the more aggressive quantization-aware LoRA framework, QA-LoRA (Xu et al., 2024), proposed for 2-bit fully quantized weight inference. QA-LoRA maintains the quantization state after merging quantized pre-trained weights (W_q) with FP16 adapters (AB), updating the adapter weights as the zero-point quantization parameter (z) in Eq. 1. Consequently, QA-LoRA employs a zero-initialization method for adapters.[5]

We apply our adaptive rank adjustment method to QA-LoRA to enhance both accuracy and efficiency in fully quantized inference. Fine-tuning experiments on LLaMA-1-7B and LLaMA-2-7B models using the Commonsense QA (CSQA) dataset show that our rank adaptive approach (RA-LoRA) consistently outperforms the baseline QA-LoRA scenario with 2-bit fully quantized weights, achieving higher accuracy with fewer parameters, as detailed in Table 4. Additionally, Table 3 demonstrates RA-LoRA's effectiveness on the GSM8K

[5]Since adapter components A and B have differing dimensions, applying LoftQ SVD-based quantization error initialization to QA-LoRA adapters is not intuitive.

Method	Bit	Rank	Trainable (%) ↓	LLaMA-1-7B CSQA Accuracy (%) ↑					LLaMA-2-7B CSQA Accuracy (%) ↑				
				PIQA	Hella.	ARC-E	ARC-C	Avg.	PIQA	Hella.	ARC-E	ARC-C	Avg.
Pre-Trained	16	N/A	N/A	78.02	56.92	75.29	41.81	63.01	78.56	57.14	76.30	43.34	63.84
QA-LoRA	2	16	0.59	75.63	48.42	66.54	35.15	56.44	73.83	49.75	65.53	34.56	55.92
RA-LoRA	2	16	0.44	76.03	48.72	66.61	35.75	**56.78**	74.54	50.56	66.33	36.18	**56.90**
QA-LoRA	2	64	2.32	75.79	48.82	66.50	35.75	56.72	74.21	50.75	66.75	35.67	56.85
RA-LoRA	2	64	1.72	76.33	49.24	66.71	36.52	**57.20**	75.73	51.05	66.71	36.43	**57.48**
QA-LoRA	2	256	8.67	76.12	48.92	62.27	36.60	57.23	74.88	51.02	67.26	35.84	57.25
RA-LoRA	2	256	6.54	76.63	50.96	67.59	36.86	**58.01**	75.99	51.24	67.20	38.05	**58.12**

Table 4: Results of 2-bit fully quantized QAT on commonsense QA tasks (PIQA, HellaSwag, ARC-E, and ARC-C) using LLaMA-1-7B and LLaMA-2-7B models, with RA-LoRA indicating the application of a rank-adaptive approach to QA-LoRA.

dataset with LLaMA-2-7B. Notably, RA-LoRA surpasses quantization-aware training (Full QAT), which updates all parameters.

To investigate the efficiency of 2-bit training, we integrate our implementation with the custom kernel[6] designed to efficiently handle operations with quantized weights and FP16 inputs. This kernel enables the packing of base weights for enhanced storage and computational efficiency. For fair comparison of performance, we do not incorporate additional memory optimization techniques, such as checkpointing and gradient accumulation. The benefits of this approach are clearly demonstrated in the speed and memory shown in Table 3, showcasing nearly double the speed improvement and triple the memory savings when compared to traditional QAT methods. This significant enhancement in both speed and memory usage underscores the effectiveness of RA-LoRA, highlighting its potential for practical applications in much aggressive quantization-aware LoRA framework.

Our RA-LoRA method has significantly improved the correction of quantization errors compared to alternatives like QLoRA or LoftQ. However, further boosting task accuracy for 2-bit quantized models remains challenging. To address this, we incorporated knowledge distillation (KD) techniques into LoftQ (detailed in Appendix A.4 and illustrated in Fig. 10). Although KD has greatly enhanced task adaptation across various ranks, it highlights the need for higher ranks. As future work, we propose to enhance 2-bit fine-tuning accuracy by dynamically adjusting ranks and optimizing fine-tuning loss objectives with methods like KD, aiming to approach full-precision model performance.

[6]https://github.com/AutoGPTQ/AutoGPTQ

8 Conclusion

This paper presents Rank Adaptive Low-Rank Adaptation (RA-LoRA) as an innovative solution to the high parameter and memory challenges of deploying large language models (LLMs). RA-LoRA adjusts adapter ranks through rank-subspace analysis, significantly mitigating quantization errors in low-bit scenarios and outperforming traditional techniques. Our findings reveal that RA-LoRA boosts accuracy in 2-bit fine-tuning of cutting-edge LLMs while reducing the number of necessary trainable parameters, marking a substantial leap in quantization-aware fine-tuning by emphasizing the critical role of rank subspace dynamics.

Acknowledgements

We express our gratitude to Geonho Lee for his valuable assistance with experiments. This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grangts funded by the Korea government (MSIT) (2024-RS-2023-00253914, artificial intelligence semiconductor support program to nurture the best talents, and 2020-0-01297, Development of Ultra-Low Power Deep Learning Processor Technology using Advanced Data Reuse for Edge Applications), and the Artificial Intelligence Industrial Convergence Cluster Development Project, funded by the Ministry of Science and ICT (MSIT, Korea) and Gwangju Metropolitan City. This work was also supported in part by Samsung Advanced Institute of Technology, Samsung Electronics Co., Ltd.

Limitations

This study provides an initial analysis of the high-rank nature of low-bit quantization errors and suggests investigating sublayer diversity to mitigate these errors. Although it highlights the distinct rank properties of quantization errors across sublayers, further exploration into the underlying reasons is highly appreciated. Our research, primarily oriented towards task-specific applications of LoRA, also identifies instruction fine-tuning as a promising direction for future research, which could expand the scope of quantization-aware LoRA strategies.

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online. Association for Computational Linguistics.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*.

Gholami. 2021. Ai and memory wall. *RiseLab Medium Post*.

Han Guo, Philip Greengard, Eric Xing, and Yoon Kim. 2024. LQ-loRA: Low-rank plus quantized matrix decomposition for efficient language model finetuning. In *The Twelfth International Conference on Learning Representations*.

Jihun Hamm and Daniel D. Lee. 2008. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 376–383, New York, NY, USA. Association for Computing Machinery.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Minsoo Kim, Sihwa Lee, Janghwan Lee, Sukjin Hong, Du-Seong Chang, Wonyong Sung, and Jungwook Choi. 2023. Token-scaled logit distillation for ternary weight generative language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. 2024. Squeezellm: Dense-and-sparse quantization.

Janghwan Lee, Minsoo Kim, Seungcheol Baek, Seok Hwang, Wonyong Sung, and Jungwook Choi. 2023. Enhancing computation efficiency in large language models through weight and activation quantization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14726–14739, Singapore. Association for Computational Linguistics.

Yixiao Li, Yifan Yu, Chen Liang, Nikos Karampatziakis, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2024. Loftq: LoRA-fine-tuning-aware quantization for large language models. In *The Twelfth International Conference on Learning Representations*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, Chuang Gan, and Song Han. 2023. Awq: Activation-aware weight quantization for llm compression and acceleration.

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

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

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. 2024. Omniquant: Omnidirectionally calibrated quantization for large language models. In *The Twelfth International Conference on Learning Representations*.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, XIAOPENG ZHANG, and Qi Tian. 2024. QA-loRA: Quantization-aware low-rank adaptation of large language models. In *The Twelfth International Conference on Learning Representations*.

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

Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. 2019. Residual learning without normalization via better initialization. In *International Conference on Learning Representations*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

A Appendix

A.1 Training Details

Natural Language Understanding (NLU). For fine-tuning the DeBERTa-V3 (He et al., 2021) model on the GLUE (Wang et al., 2019) tasks, we consistently used a batch size of 32 across all settings, following the LoftQ configuration. The learning rate was tailored to each task as follows: CoLA, RTE, and MRPC at 1E-4, STS-B at 5E-5.

Natural Language Generation (NLG). We fine-tuned the LLaMA-2-7b and LLaMA-2-13b (Touvron et al., 2023) models on the wikitext and GSM8K datasets. For wikitext, both models were trained with a block size of 512 and a uniform learning rate of 1E-4, regardless of the adapter rank, using a batch size of 16.

In the case of GSM8K, a batch size of 16 was employed for both models. When the target rank was 128 or below, a learning rate of 3E-4 was used, and for a target rank of 256, the learning rate was set to 1E-4.

Evaluation with QA-LoRA. For applying QA-LoRA (Xu et al., 2024) to the LLaMA-2-7b model with GSM8K fine-tuning, a batch size of 1 was used. The learning rate was adjusted based on the target rank: 3E-4 for ranks 64 and 1E-4 for a rank of 256.

RA-LoRA Implementation To implement the RA-LoRA algorithm, we used the hyper-parameters $\alpha = 64$ for $\mathcal{R}_0$ and $\mathcal{R}_1$, $\beta = 16$ for $\mathcal{R}_2$ and $\mathcal{R}_3$, and $\gamma = 1$ for $\mathcal{R}_4$, $\mathcal{R}_5$, and $\mathcal{R}_6$. Specifically, for LLaMA 3rd transformer block, following each sublayer's NCSV, we applied α to the Query and Key sublayers, β to the Value and Output sublayers, and γ to the FC1, FC2, and Gate sublayers. All experimental results in this paper were measured using this combination of hyper-parameters to evaluate performance and calculate the trainable parameter ratio. This combination of hyper-parameters can be further optimized, and devising a method to more dynamically determine the rank of sublayers according to NCSV values is highly appreciated. Our implementation is based on LoftQ (Li et al., 2024) git code and Huggingface PyTorch code base[7]. The QA-LoRA experiment is conducted based on the QA-LoRA source code[8].

[7]https://github.com/yxli2123/LoftQ
https://github.com/huggingface/transformers

[8]https://github.com/eltociear/qa-lora

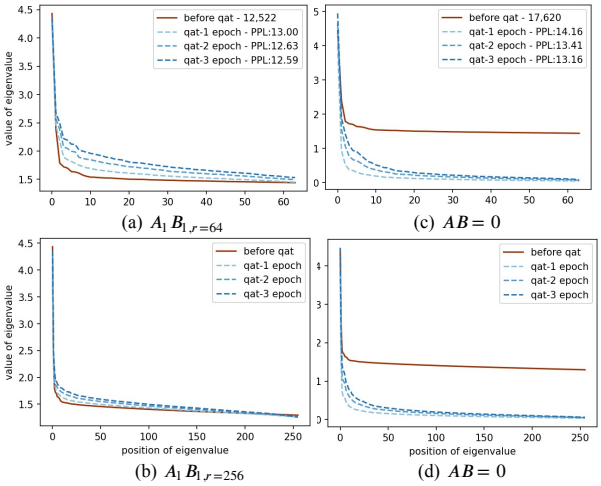


Figure 6: Singular values of low-rank adapters across fine-tuning epochs with PPL performance. (a),(b) fine-tuning LoftQ-1step initialization. rank 64 and 256 respectively (c),(d) $AB = 0$ rank 64 and 256 respectively. (OPT-1.3B 15th Query layer with Wikitext fine-tuning)

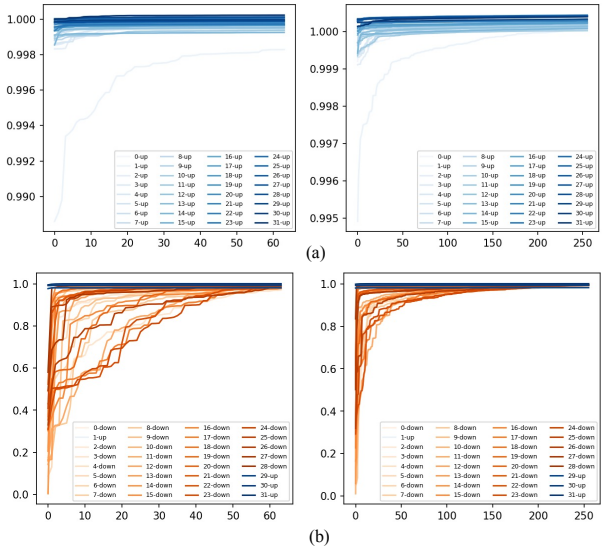


Figure 7: Normalized similarity between quantization error and subspaces spanned by singular vectors of learned adapter per layer. (a) Query projection (b) Up protection, Left: $U_{A_{r=64}B_{r=64}}$ Right: $U_{A_{r=256}B_{r=256}}$ LLaMA-2-7b wikitext fine-tuning.

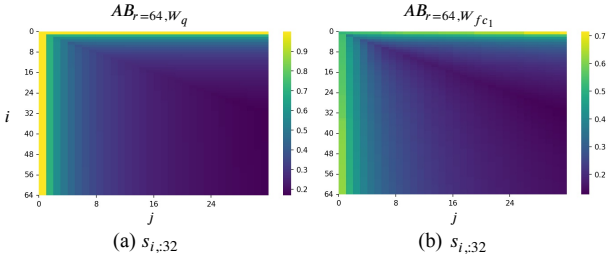


Figure 8: Similarity distance heatmap between W_{qerr} and $AB_{r=64}$. (a) Query layer (b) FC1 layer. To visualize the differences in color for each element of the heatmap, we truncate the dimension along the j axis to 32.

A.2 Supplementary Data for Main Analysis

Singular Values Evolution from Fine-Tuning We extend the analysis of the evolution of adapter matrix singular values during fine-tuning, as observed in Fig. 2, to the OPT-1.3B model (Zhang et al., 2022) undergoing fine-tuning on Wikitext. Consistent with observations made in Sec. 5.1, we find that with quantization error initialization, an increase in singular values is evident across all positions for both rank 64 and 256. In contrast, with zero initialization, an increase in singular values is observed only in the few singular values, while the remaining positions exhibit values close to zero.

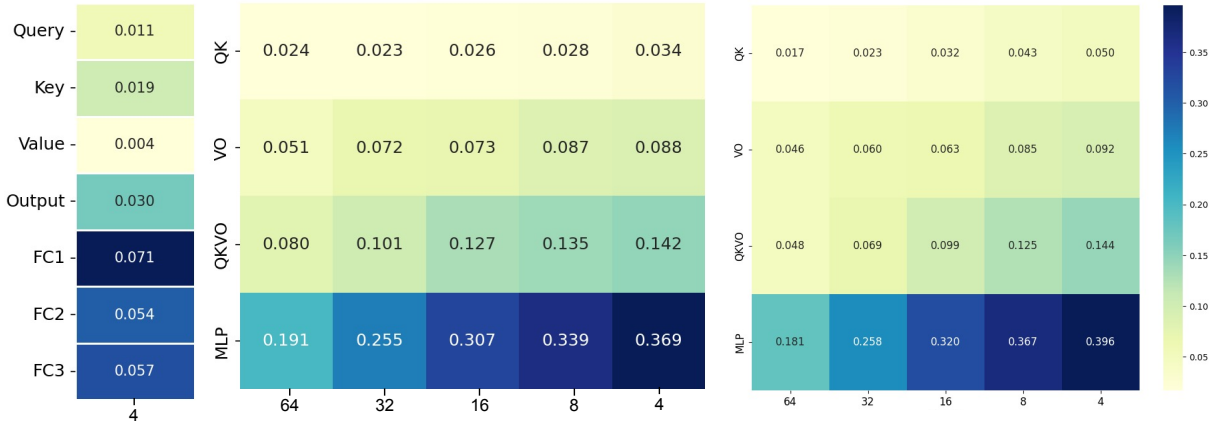
Subspace Similarity with Quantization Error In Sec. 5.2, we validate the learning adapters subspace similarity patterns in wikitext fine-tuning. Similar to the findings presented in Fig. 3, high similarity with quantization error in the Query layer could be achieved with low ranks, while in the FC layer, similarity increased progressively with the expansion of the singular vector dimension spanned as shown in Fig. 7.

A notable distinction was observed in the dimension at which similarity in the Query layer converged, occurring much earlier than depicted in Fig. 3. This phenomenon likely reflects the differing characteristics of the fine-tuning tasks. Given that GSM8K (Cobbe et al., 2021) involves solving multi-step mathematical problems, which is inherently generation-based, it suggests a higher degree of adapter learning might be required compared to the language modeling task of Wikitext (Merity et al., 2016). Such differences suggest that the rank needed for similarity convergence in the Query layer varies, implying that the *effective rank may differ based on the complexity of the fine-tuning task*. This variation underscores the notion that the difficulty of the fine-tuning task can influence the optimal rank for achieving effective learning outcomes.

Subspace Similarity between Different Rank

$$s_{i,j} = \phi(W_{qerr}, AB_r, i, j) = \frac{||U^{i\top}_{W_{qerr}} U^{j}_{AB_r}||^2_F}{\min(i,j)} \in [0,1] \quad (6)$$

In this study, we employed two orthogonal matrices to quantify the similarity between two subspaces. These matrices utilize the right singular



(a) (b)

Figure 9: Fine-tuning performance ablation study across various ranks. Each cell represents the performance degradation when adjusting the rank of specific sub-layers, as indicated on the x-axis during LLaMA-2-7B Fine-Tuning on WikiText-2 with rank 256. (a) LoftQ (Li et al., 2024) ($r = 256$ PPL: 6.8427) (b) LQ-LoRA (Guo et al., 2024) ($r = 256$ PPL: 6.6939)

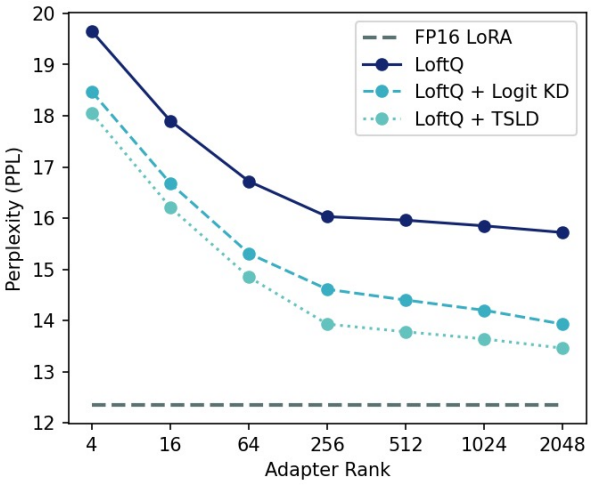


Figure 10: OPT-1.3b WikiText-2 LoftQ PPL results with Knowledge Distillation methods with adapter rank exploration. Logit KD denotes conventional logit distillation and TSLD denotes Token-Scaled Logit Distillation (Kim et al., 2023) KD method for decoder-only model.

unitary values from the SVD results of the quantization error matrix (W_{qerr}) and the learned adapter matrix (AB_r) with rank r respectively. Our goal in this analysis was to gauge the degree of similarity between the subspace spanned by the top i singular vectors and the corresponding subspace spanned by the quantization error as described in Eq. 6. Through this analysis, we aimed to observe the extent to which the subspace of the learned adapter matrix approximates the quantization error, thereby providing insights into how effectively the learned adapter can mitigate quantization errors.

In this analysis, we measure the similarity between the learned adapter and the subspace formed by the top singular vectors of the quantization error. This approach was chosen because, in the context of strong 2-bit quantization errors, vectors beyond the top-K singular vectors tend to exhibit a random noise characteristic, making it challenging to derive meaningful similarity values. As illustrated in Fig. 8, an increase in similarity is observed as the dimension of j increases when i is 0, a trend consistent with what is observed in Fig. 3 and Fig. 8. A key insight from this analysis is that there is a variation in the degree to which similarity with the quantization error subspace is restored, depending on the type of sublayer as discussed in Sec. 5.2.

A.3 Rank Ablation Study

Another perspective to observe the diversity of effective ranks across sublayers is through a rank ablation study. Fig. 9 explores the performance degradation when assigning specific low ranks, ranging from 4 to 64, to particular sublayers across all layers, while keeping the overall rank fixed at 256. From Fig. 9(a), it is evident that the MLP layers (FC1 - 3) exhibit a higher sensitivity to performance degradation than the self-attention layer. This finding aligns with discussions in Sec. 5.2 and 6.1 about the FC layer demonstrating characteristics requiring a high rank.

In the case of Fig. 9(b), the results reflect an exploration using the LQ-LoRA (Guo et al., 2024) method, which incorporates Fisher information in the process of initializing the adapter with quanti-

zation errors. This approach, considering task loss through SVD initialization, shows an improvement in performance from 6.8426 to 6.6939 with an overall rank of 256. Notably, the pattern of sensitivity observed across sublayers remains consistent even with the application of Fisher information. Thus, through rank exploration, the diversity in rank characteristics among sublayers can be reaffirmed.

A.4 Rank Exploration with KD

An additional intriguing aspect when exploring the properties of adapter rank in the context of low-bit quantization errors is the impact of Knowledge Distillation (KD) (Hinton et al., 2015). We sought to investigate whether incorporating KD into adapter learning could effectively tackle the adverse effects of high-rank quantization errors.

In our exploration with the OPT-1.3b model (Zhang et al., 2022) using the LoftQ method for rank exploration, we examine how fine-tuning performance changes when adding logit distillation to the loss objective or Token-Scaled Logit Distillation (TSLD) (Kim et al., 2023), which facilitates better learning from the teacher model in a decoder-only model.

The results reveal that logit distillation noticeably improves fine-tuning PPL performance across all ranks, with TSLD further enhancing PPL. However, a critical observation is that a significantly high rank is still required to approach full precision performance. Thus, these experiments suggest that by leveraging the performance-boosting effects of logit distillation while handling high-rank characteristics through rank-adaptive methods, it is possible to bring the performance of 2-bit quantization closer to FP levels. We will leave these efforts for future work.