CoCaQ: Coupling of Compression and Quantization for Flexible LLM Deployment

Anonymous authors
Paper under double-blind review

Abstract

The practical deployment of Large Language Models (LLMs), especially on edge devices, is heavily constrained by their massive model sizes and prohibitive computation costs, which are generally addressed by either model compression or quantization. In practice, these two kinds of techniques can be applied concurrently for flexible LLM deployment, especially under extremely low memory budgets, which, however, are typically optimized separately and so lead to suboptimal performance. Particularly, we observe that the joint of model compression and quantization is challenged by the effect of compressed structures on quantization and the ability of quantization to account for distribution shifts induced by compression. To address the above issues, we propose a new Coupling of Compression and Quantization (**CoCaQ**) method to perform model compression and quantization in a coupling manner, aiming to achieve effective and flexible LLM deployment by unifying the two techniques. To this end, our CoCaQ is developed with two key components, including a quantization-aware model compression (QaMC) and an activation stabilization for quantization (ASQ). First, our QaMC constructs quantization-friendly low-rank representations based on activation statistics and optimizes the layer-wise compressed structures with a quantization-aware loss, aiming to identify the robust substructures of LLM for the subsequent quantization. Furthermore, our ASQ introduces an efficient orthogonal rotation regularization for low-rank weight matrices, enabling low-bit quantization to better tolerate activation amplification induced by the preceding model compression. Based on the proposed QaMC and ASQ, our CoCaQ couples model compression and quantization by fully considering the mutual influence between the two, resulting in an effective yet flexible solution for LLM deployment. Experiments are conducted across LLaMA and Qwen models, and the results show that our CoCaQ achieves better memory-accuracy trade-offs than state-of-the-art compression-only and quantization-only methods, while clearly outperforming the counterparts with joint of compression and quantization.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success in natural language understanding (Brown et al., 2020; Ouyang et al., 2022), reasoning (Wei et al., 2022; Kojima et al., 2022), coding (Chen et al., 2021; Nijkamp et al., 2022), and multimodal applications (Liu et al., 2023; Wang et al., 2026c). However, their rapidly growing scale leads to high memory consumption and computational costs during inference, making deployment difficult on resource-constrained platforms such as edge devices, consumer GPUs, and latency-sensitive serving systems. To address these challenges, many techniques have been explored, including low-rank weight decomposition (Yu & Wu, 2023; Chang et al., 2024; Wang et al., 2025; Thoma et al., 2026) and low-bit quantization (Xiao et al., 2023; Lin et al., 2024; Ashkboos et al., 2024; Liu et al., 2025; Park et al., 2026). Among them, low-rank weight decomposition compresses models by factorizing weight matrices into low-rank components, but large reduction rates will severely limit the expressive capacity of the weights, resulting in large reconstruction errors and substantial performance degradation, as shown in Fig. 1. Differently, low-bit quantization represents weights and activations with low-bit formats to reduce model size. However, as shown in Fig. 1, its discrete bit-widths limit deployment flexibility, especially ultra low-bit quantization (e.g., W2A8) can cause a sharp performance drop, leaving the whitespace in memory requriement (e.g., 2.5GB$\sim$3GB for LLaMA-2-7B). Thus, current compression-only or

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

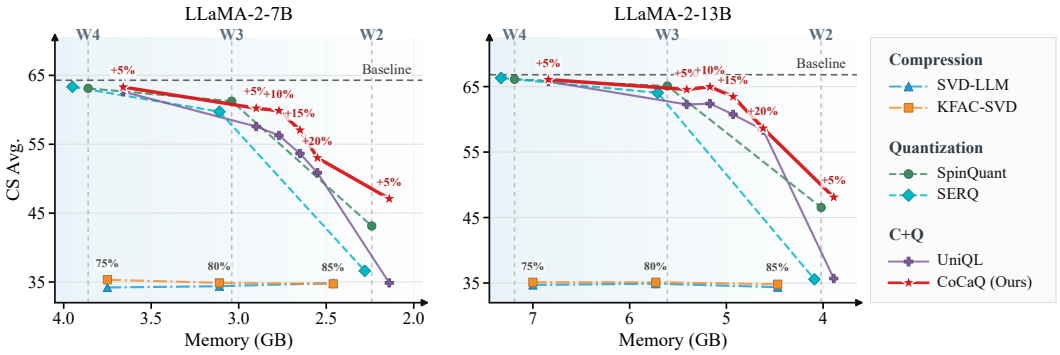


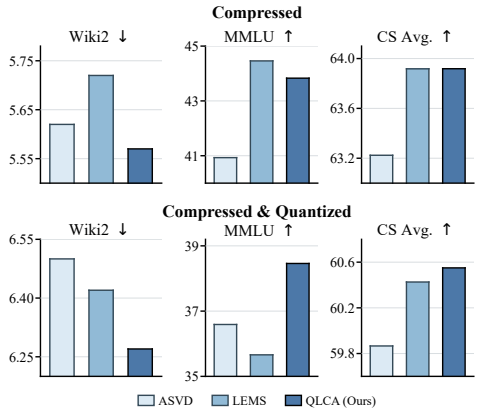
Figure 1: Performance comparison of compression-only, quantization-only, and compression–quantization coupling (C+Q) methods under different memory budgets on LLaMA-2-7B and LLaMA-2-13B. The settings correspond to W4A8, W3A8, and W2A8-level memory budgets. Compared with FlatQuant (Sun et al., 2025), SERQ (Park et al., 2026), SVD-LLM (Wang et al., 2025), KFAC-SVD (Thoma et al., 2026), and UniQL (Chiang et al., 2026), CoCaQ achieves a superior performance–memory trade-off with better flexibility to memory budgets, especially in the W3–W2 settings, where it flexibly adapts to different memory budgets while maintaining strong performance.

quantization-only methods hardly provide sufficient flexibility and performance stability under low memory budgets.

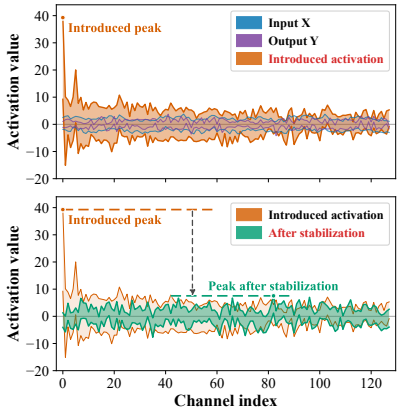
Consequently, the combination of the two techniques potentially enables better accuracy-efficiency trade-offs, which mitigates degradation from either technique alone by distributing the memory costs, and so supporting flexible deployment under diverse memory budgets. Previous studies (Yuan et al., 2023; Li et al., 2024; Xv et al., 2025) have attempted to combine them sequentially, treating compression and quantization as independent stages. UniQL (Chiang et al., 2026) further integrates quantization into SVD by modifying the singular-value distribution between low-rank factors, but applies this approach only to two specific types of layers. Despite these efforts, existing methods still do not fully consider the mutual influence between compression and quantization, resulting in sub-optimal performance. Particularly, we observe that effective joint of model compression and low-bit quantization is challenged by two issues. First, as illustrated in Fig. 2a, the compressed structures of better performance don't always hold after low-bit quantization, implying post quantization is sensitive to the preceding compressed structures, where inappropriate structures will significantly degrade performance after low-bit quantization. Second, as shown at the top of Fig. 2b, low-rank weight decomposition will introduce extra outliers absent from the original distribution, thereby shifting the activation distribution. They potentially increase quantization errors due to the enlargement of activations range.

To address these challenges, we propose **CoCaQ**, a new method to effectively couple low-rank weight decomposition and low-bit quantization for flexible yet efficient LLM deployment. Specifically, our CoCaQ presents a quantization-aware model compression (QaMC) to address the structure sensitivity of post-quantization performance, while activation stabilization for quantization (ASQ) is introduced to mitigate compression-induced shifts in activation distributions. To construct compressed models that are robust to low-bit quantization, QaMC first employs quantization-friendly singular value decomposition (QF-SVD) to construct low-rank representations guided by activation statistics. Building upon these quantization-friendly representations, it then applies quantization-aware layer-wise compression allocation (QLCA) to select compressed structures under a given memory budget based on their post-quantization robustness. To mitigate the activation distribution shifts caused by low-rank weight decomposition, our ASQ develops an orthogonal rotation regularization to stabilize the resulting activation distributions, which restores a more quantization-friendly dynamic range and thereby reduces the quantization errors induced by the newly introduced outliers. By accounting for the mutual influence between low-rank weight decomposition and quantization, CoCaQ achieves superior accuracy-memory trade-offs especially under tight memory budgets while offering better memory flexibility. To evaluate the effectiveness of our CoCaQ, experiments are conducted on LLaMA-2 (Touvron et al., 2023) models (7B/13B), LLaMA-3 (Grattafiori et al., 2024) 8B, and Qwen3 (Yang et al., 2025) models (8B/14B).

Our contributions can be summarized as follows:



(a) Impact of different compressed structures obtained by ASVD (Yuan et al., 2023), LEMS (Thoma et al., 2026), and QLCA (Ours) on subsequent quantization.



(b) **Top**: low-rank weight decomposition introduces extra activation outliers, impacting subsequent quantization. **Bottom**: our ASQ effectively stabilizes the outliers.

Figure 2: Illustration of challenges lying in unifying model compression and low-bit quantization.

- This paper first analyzes the key influence lying in unifying model compression and quantization, and then proposes a training-free Coupling of Compression and Quantization (CoCaQ) method to achieve flexible and efficient LLM deployment with effective mix of both two techniques.
- To this end, CoCaQ proposes a QaMC to identify robust compressed structures with quantization-friendly low-rank representations and an ASQ to stabilize the resulting activation distributions, which effectively address the challenges posed by the effects of compressed structures on quantization and compression-induced activation shifts, respectively.
- Extensive experiments show that our CoCaQ achieves superior performance over state-of-the-art compression-only methods, and offers better memory flexibility than state-of-the-art quantization-only methods. Compared with existing joint compression-and-quantization methods, CoCaQ achieves clear gains especially under low memory budgets.

2 RELATED WORK

Low-rank weight decomposition (Denton et al., 2014; Ganesh et al., 2021) compresses weight matrices with low-rank factors and has been widely applied to LLMs (Yuan et al., 2023; Wang et al., 2025; Thoma et al., 2026). Specifically, ASVD (Yuan et al., 2023) introduces activation-aware transformations to preserve important directions during decomposition, while SVD-LLM (Wang et al., 2025) employs whitening and truncation-aware optimization to reduce the approximation error of the resulting low-rank factors. KFAC-SVD (Thoma et al., 2026) improves the decomposition quality by incorporating token-wise statistics to avoid rank deficiency. However, large rank reduction under low memory budgets retains little information, which limits the capacity of the decomposed weights, and so leads to substantial performance degradation.

Low-bit quantization reduces the memory footprint of LLMs by representing weights and activations in low-bit formats. Recent methods (Lin et al., 2024; Ashkboos et al., 2024; Xiang & Zhang, 2024; Liang et al., 2026; Park et al., 2026) improve low-bit accuracy by mitigating activation outliers, e.g., through weight-activation rescaling in SmoothQuant (Xiao et al., 2023) or orthogonal rotations in SpinQuant (Liu et al., 2025). FlatQuant (Sun et al., 2025) learns layer-wise affine transformations to flatten weight and activation distributions, improving quantization performance. SERQ (Park et al., 2026) employs saliency-aware low-rank error reconstruction with a single compensation matrix to mitigate quantization errors from both activation and weight saliency. However, quantization relies on discrete bit-widths, which limits the granularity of achievable memory footprints. Under tight memory budgets, this often requires very low bit-widths, where limited numerical representation capacity causes substantial information loss and severe performance degradation.

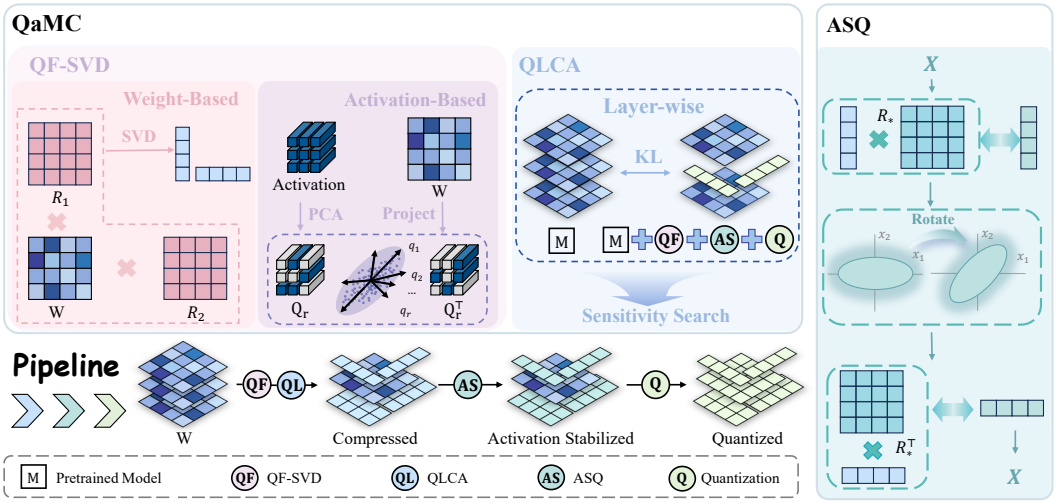


Figure 3: Overview of the proposed CoCaQ method. Specifically, CoCaQ first presents QaMC to respectively construct quantization-friendly low-rank representations (i.e., QF-SVD) and determine the compressed model structure (i.e., QLCA), and then ASQ stabilizes intermediate activations for the subsequent low-bit quantization.

Joint compression and quantization has recently attracted increasing attention, with several works (Li et al., 2024; Xv et al., 2025; Wang et al., 2026c;a;b) applying quantization to models compressed by low-rank decomposition. However, these approaches typically treat the two stages independently, applying quantization only after compression without considering their mutual influence. UniQL (Chiang et al., 2026) takes a further step by modifying the distribution of singular values between low-rank factors to improve quantization compatibility, but applies this strategy only to two specific types of layers. Nevertheless, existing methods have not systematically explored the challenges arising from jointly applying compression and quantization. Different from previous methods, we propose CoCaQ by fully considering the mutual influence between the two techniques, achieving clear performance gains over existing approaches, as shown in Table 2.

3 METHOD

3.1 OBSERVATIONS AND OVERALL FRAMEWORK

To effectively unify model compression and low-bit quantization, we first carefully investigate their mutual influence during the joint working from two perspectives, including the sensitivity of low-bit quantization to the preceding compressed structures and the impact of compression-induced abnormal activations on the subsequent low-bit quantization.

As illustrated in Fig. 2a, we apply the same SVD-based decomposition method with different compressed structures under the same memory budget, comparing their performance before and after quantization. Particularly, the compressed structures of better performance don't always hold after low-bit quantization. For example, the structure determined by LEMS (Thoma et al., 2026) achieves better performance than ASVD (Yuan et al., 2023) and our QLCA on MMLU and commonsense reasoning benchmarks before quantization, but suffers a larger performance degradation after low-bit quantization while underperforming both ASVD and QLCA on MMLU. These results clearly demonstrate that the compressed structure has a significant impact on subsequent quantization, and better reconstruction quality cannot easily transform into better quantization performance. It suggests that the influence of the compressed structure should be considered in subsequent quantization.

As illustrated in Fig. 2b, compression introduces an extra intermediate activation between the two low-rank factors. Particularly, we observe that the distribution of these intermediate activations is shifted relative to the input and output activations of the corresponding layer, and introduces additional outliers. These outliers enlarge the activation dynamic range and consequently reduce the

Algorithm 1 CoCaQ: Coupling of Compression and Quantization

Input: Pre-trained model M, calibration set D, memory budget B_{target}
Output: Compressed and quantized model M_q
Derive the compression ratio ρ and quantization bit-width b from B_{target}
Generate layer-wise candidate ranks set of $\mathbb{R}^l = \{r_1^l, \ldots, r_k^l\}$ under ρ
For each layer l and candidate rank $r \in \mathbb{R}^l$ **do**
    Compute QF-SVD factors of $\boldsymbol{W}^l \approx \widehat{\boldsymbol{A}_r^l}\widehat{\boldsymbol{B}_r^l}$
    Fold the orthogonal transform into the factors: $\tilde{\boldsymbol{A}}_r^l = \widehat{\boldsymbol{A}_r^l}\boldsymbol{R}_*^l$, $\tilde{\boldsymbol{B}}_r^l = (\boldsymbol{R}_*^l)^\top \widehat{\boldsymbol{B}_r^l}$
    Quantize the low-rank layer and estimate sensitivity
    $\mathcal{L}_{\text{Q}}(l, r, b) = D_{\text{KL}}\Big(P(M) \,\|\, P(\text{Q}(\tilde{\boldsymbol{A}}_r^l \tilde{\boldsymbol{B}}_r^l, b))\Big)$
End For
Select ranks $\{r^l\}_{l=1}^L$ by a binary search under $\sum_{l=1}^L \text{B}(l, r, b) \le B_{\text{target}}$
Apply QF-SVD with $\{r^l\}_{l=1}^L$ and fold $\boldsymbol{R}_*^l$ into each compressed layer
Quantize the resulting model with bit-width b to obtain M_q
Return M_q

effective numerical precision under low-bit representations, which can significantly affect activation quantization (Dettmers et al., 2022; Xiao et al., 2023; Liu et al., 2025; Liang et al., 2026).

Based on these observations, we propose **CoCaQ** to couple low-rank decomposition and quantization, as illustrated in Fig. 3. Given a target memory budget, CoCaQ first uses **QaMC** to construct a quantization-friendly compressed model and then applies **ASQ** to stabilize intermediate activations. The details of QaMC and ASQ are presented in Sec. 3.2 and Sec. 3.3, respectively.

3.2 QaMC: quantization-aware model compression

Our QaMC consists of QF-SVD 3.2.1 and QLCA 3.2.2. QF-SVD derives quantization-friendly low-rank representations by exploiting activation statistics and reparameterizing the factors, while QLCA determines the layer-wise compression allocation according to the post-quantization performance of candidate compressed structures. Together, they establish a compression process to account for its effect on subsequent quantization, yielding a quantization-aware compressed model.

3.2.1 QF-SVD: Quantization-Friendly Singular Value Decomposition

Generally, low-rank weight decomposition and low-bit quantization introduce two distinct sources of error. Let $\boldsymbol{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, $\boldsymbol{X} \in \mathbb{R}^{d_{\text{in}} \times N}$, and $\boldsymbol{A}_r \in \mathbb{R}^{d_{\text{out}} \times d_{\text{r}}}$, $\boldsymbol{B}_r \in \mathbb{R}^{d_{\text{r}} \times d_{\text{in}}}$ stand for the low-rank factors after decomposition. We have

$$\mathcal{L}_{\text{rec}} = \|\boldsymbol{W}\boldsymbol{X} - \boldsymbol{A}_r\boldsymbol{B}_r\boldsymbol{X}\|_{\text{F}}, \qquad \mathcal{L}_{\text{quant}} = \|\boldsymbol{A}_r\boldsymbol{B}_r\boldsymbol{X} - \text{Q}(\boldsymbol{A}_r)\,\text{Q}(\boldsymbol{B}_r)\boldsymbol{X}\|_{\text{F}}, \tag{1}$$

where $\text{Q}(\cdot)$ denotes quantization. As such, the total model error can be computed as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{quant}} = \|\boldsymbol{W}\boldsymbol{X} - \boldsymbol{A}_r\boldsymbol{B}_r\boldsymbol{X}\|_{\text{F}} + \|\boldsymbol{A}_r\boldsymbol{B}_r\boldsymbol{X} - \text{Q}(\boldsymbol{A}_r)\,\text{Q}(\boldsymbol{B}_r)\boldsymbol{X}\|_{\text{F}}, \tag{2}$$

where $\mathcal{L}_{\text{total}}$ jointly captures the reconstruction and quantization errors. Given the original weight matrix $\boldsymbol{W}$, we introduce an orthogonal reparameterization, followed by SVD truncation to the leading r components, as

$$\boldsymbol{R}_{out}\boldsymbol{W}\boldsymbol{R}_{in} \approx \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top = (\boldsymbol{U}_r\boldsymbol{\Sigma}_r^{\frac{1}{2}})(\boldsymbol{\Sigma}_r^{\frac{1}{2}}\boldsymbol{V}_r^\top) = \boldsymbol{A}_r\boldsymbol{B}_r, \tag{3}$$

where $\boldsymbol{R}_{in}$ and $\boldsymbol{R}_{out}$ are random Hadamard matrices for the input and output. Since $\boldsymbol{R}^{-1} = \boldsymbol{R}^\top$, the rotations preserve the original input-output mapping and leave $\mathcal{L}_{\text{rec}}$ unchanged, which is proved in the Appendix A.1. Meanwhile, they smooth the numerical distributions of the SVD factors, thereby reducing $\mathcal{L}_{\text{quant}}$, as illustrated in the Appendix A.2. As such, our Eq. (3) provides a quantization-friendly SVD. However, its effectiveness degrades with increasing compression ratios as reconstruction error grows.

To reduce reconstruction error with a large compression ratio, we further introduce an activation-based low-rank weight decomposition strategy, which decomposes weights by projecting them onto

the feature subspace. Specifically, given the output activations $\boldsymbol{Y}$ on a small calibration set, we can compute the covariance matrix of $\boldsymbol{Y}$ as

$$\boldsymbol{C} = \boldsymbol{Y}\boldsymbol{Y}^\top = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top, \tag{4}$$

and then we project $\boldsymbol{W}$ into the subspace with top-r components, i.e.,

$$\boldsymbol{W} = \boldsymbol{I}\boldsymbol{W} \approx \boldsymbol{Q}_r(\boldsymbol{Q}_r^\top \boldsymbol{W}) = \boldsymbol{A}_r\boldsymbol{B}_r. \tag{5}$$

Thus the reconstruction error $\mathcal{L}_{\text{rec}}$ can be reformulated as

$$\begin{aligned}\mathcal{L}_{\text{rec}} &= \|\boldsymbol{W}\boldsymbol{X} - \boldsymbol{A}_r\boldsymbol{B}_r\boldsymbol{X}\|_F = \left\|\boldsymbol{W}\boldsymbol{X} - \boldsymbol{Q}_r\boldsymbol{Q}_r^\top\boldsymbol{W}\boldsymbol{X}\right\|_F = \left\|\boldsymbol{Y} - \boldsymbol{Q}_r\boldsymbol{Q}_r^\top\boldsymbol{Y}\right\|_F \\ &= \left\|(\boldsymbol{I} - \boldsymbol{Q}_r\boldsymbol{Q}_r^\top)\boldsymbol{Y}\right\|_F = \mathrm{tr}\big(\boldsymbol{Y}^\top(\boldsymbol{I} - \boldsymbol{Q}_r\boldsymbol{Q}_r^\top)\boldsymbol{Y}\big)^{\frac{1}{2}},\end{aligned} \tag{6}$$

where $\mathrm{tr}(\cdot)$ denotes the matrix trace. Let $\boldsymbol{Q} = [\boldsymbol{Q}_r, \boldsymbol{Q}_\perp] \in \mathbb{R}^{d_{out}\times d_{out}}$, we have $\boldsymbol{I} - \boldsymbol{Q}_r\boldsymbol{Q}_r^\top = \boldsymbol{Q}_\perp\boldsymbol{Q}_\perp^\top$, as $\boldsymbol{Q}$ is orthogonal. Thus the square of the reconstruction loss $\mathcal{L}_{\text{rec}}$ can be calculated as

$$\begin{aligned}{\mathcal{L}_{\text{rec}}}^2 &= \mathrm{tr}(\boldsymbol{Y}^\top(\boldsymbol{I} - \boldsymbol{Q}_r\boldsymbol{Q}_r^\top)\boldsymbol{Y}) = \mathrm{tr}(\boldsymbol{Y}^\top\boldsymbol{Q}_\perp\boldsymbol{Q}_\perp^\top\boldsymbol{Y}) = \mathrm{tr}(\boldsymbol{Q}_\perp^\top\boldsymbol{Q}\Lambda\boldsymbol{Q}^\top\boldsymbol{Q}_\perp) \\ &= \mathrm{tr}\left([0 \quad \boldsymbol{I}_{m-r}]\begin{bmatrix}\Lambda_r & 0\\ 0 & \Lambda_\perp\end{bmatrix}\begin{bmatrix}0\\ \boldsymbol{I}_{m-r}\end{bmatrix}\right) = \mathrm{tr}(\Lambda_\perp) = \sum_{j=r+1}^{d_{out}}\lambda_j,\end{aligned} \tag{7}$$

where λ_j are the eigenvalues of $\boldsymbol{C}$. Based on the Eckart–Young–Mirsky theorem (Eckart & Young, 1936), the remaining top r principal directions can minimize the rank-r reconstruction error.

Generally, the resulting factors exhibit imbalanced magnitudes $\|\boldsymbol{A}_r\|_F = \|\boldsymbol{Q}_r\|_F \neq \|\boldsymbol{B}_r\|_F = \|\boldsymbol{Q}_r^\top\boldsymbol{W}\|_F$. In particular, $\boldsymbol{B}_r$ and its corresponding activations exhibit higher energy and wider numerical ranges, as shown in the Appendix A.2, making them more challenging to quantize. We therefore apply an untruncated SVD to $\boldsymbol{B}_r$ to redistribute the energy between the two factors, i.e.,

$$\boldsymbol{A}_r\boldsymbol{B}_r = \boldsymbol{Q}_r(\boldsymbol{Q}_r^\top\boldsymbol{W}) = \boldsymbol{Q}_r(\boldsymbol{U}_b\boldsymbol{\Sigma}_b\boldsymbol{V}_b^\top) = (\boldsymbol{Q}_r\boldsymbol{U}_b\boldsymbol{\Sigma}_b^{\frac{1}{2}})(\boldsymbol{\Sigma}_b^{\frac{1}{2}}\boldsymbol{V}_b^T) = \widehat{\boldsymbol{A}_r}\widehat{\boldsymbol{B}_r} \tag{8}$$

Thus, we have $||\widehat{\boldsymbol{A}_r}||_F^2 = ||\widehat{\boldsymbol{B}_r}||_F^2 = \mathrm{tr}(\boldsymbol{\Sigma}_b)$. As illustrated in the Appendix A.2, the reparameterized factors exhibit more balanced weight and activation distributions with smaller dynamic ranges, reducing $\mathcal{L}_{\text{quant}}$ and consequently $\mathcal{L}_{\text{total}}$.

3.2.2 QLCA: Quantization-Aware Layer-wise Compression Allocation

Although QF-SVD realizes quantization-friendly low-rank weight decomposition, the layer-wise allocation of compression ratios remains challenging. Existing methods typically determine the truncation rank of each layer according to reconstruction errors or compression sensitivity. However, such criteria cannot accurately reflect the behavior of compressed models after quantization. To address this issue, we propose a quantization-aware compression allocation strategy that explicitly evaluates candidate compression structures under the target quantization configuration.

Specifically, for the l-th layer, we first apply QF-SVD in Eq. (8) with a candidate rank r, yielding a compressed model denoted by M_r^l, where M is the original model. The compressed model is then quantized using the target quantization configuration

$$\mathrm{Q}(M_r^l, b) = \mathrm{Q}(\widehat{\boldsymbol{A}_r^l}\widehat{\boldsymbol{B}_r^l}, b), \tag{9}$$

where $\mathrm{Q}(\cdot, b)$ denotes quantization with b-bit precision. The quantization-aware sensitivity of M_r^l is then measured by the KL divergence between the output token distributions of M and $\mathrm{Q}(M_r^l, b)$ as

$$\mathcal{L}_{\mathrm{Q}}(l, r, b) = D_{\mathrm{KL}}\big(\mathrm{P}(M)\|\,\mathrm{P}(\mathrm{Q}(M_r^l, b))\big) = D_{\mathrm{KL}}\Big(\mathrm{P}(M)\|\,\mathrm{P}(\mathrm{Q}(\widehat{\boldsymbol{A}_r^l}\widehat{\boldsymbol{B}_r^l}, b))\Big) \tag{10}$$

where $\mathrm{P}(\cdot)$ denotes the output token distribution of a model. By evaluating different candidate ranks, we construct a quantization-aware sensitivity mapping for each layer. Given a target parameter budget B_{target}, the optimal layer-wise rank allocation is formulated as

$$\min \sum_{l=1}^{L}\mathcal{L}_Q(l, r, b), \quad \text{s.t.} \quad \sum_{l=1}^{L}\mathrm{B}(l, r, b) \leq B_{\text{target}}, \tag{11}$$

where $\mathrm{B}(l, r, b)$ denotes the memory budget for assigning rank r to layer l and quantizing it with b bits. We solve Eq. (11) using a binary search strategy, whose details are provided in the Appendix A.5.

Table 1: Comparison of different methods on commonsense reasoning benchmarks in terms of average accuracy and memory budgets. The Setting column specifies the compression ratio and the quantization manner. **Bold** indicates the best within each group and model, while underlined denotes the second-best. C, Q, and C+Q denote compression-only, quantization-only, and joint compression with quantization, respectively. Detailed results are listed in the Appendix A.10.

Type	Method	Setting	Memory (GB) ↓					CS Avg. ↑				
			L2-7B	L2-13B	L3-8B	Q3-8B	Q3-14B	L2-7B	L2-13B	L3-8B	Q3-8B	Q3-14B
-	Baseline	FP	13.50	26.00	16.10	16.40	29.50	64.29	66.81	67.54	68.62	71.38
C	SVD-LLM (Wang et al., 2025)	75%	3.75	7.00	5.59	5.96	9.72	34.21	34.70	38.12	34.87	34.70
C	KFAC-SVD* (Thoma et al., 2026)	75%	3.75	7.00	5.59	5.96	9.72	35.33	35.12	35.31	36.01	35.30
Q	SpinQuant (Liu et al., 2025)	W4A8	3.86	7.19	5.69	6.06	9.92	63.11	66.11	67.04	68.56	**71.11**
Q	FlatQuant (Sun et al., 2025)	W4A8	3.86	7.19	5.69	6.06	9.92	63.41	66.01	66.45	67.99	70.97
Q	SERQ (Park et al., 2026)	W4A8	3.95	7.33	5.79	6.17	10.05	63.34	**66.31**	66.26	67.44	69.81
C+Q	**CoCaQ (Eq. (3))**	5%+W4A8	3.66	6.84	5.53	5.92	9.58	63.39	65.98	**67.12**	68.75	70.75
C+Q	**CoCaQ (Eq. (3))**	10%+W4A8	**3.49**	**6.61**	**5.32**	**5.71**	**9.25**	62.59	65.79	59.72	67.00	68.94
C+Q	**CoCaQ (Eq. (8))**	5%+W4A8	3.66	6.84	5.53	5.92	9.58	63.30	66.07	66.96	**68.77**	70.03
C+Q	**CoCaQ (Eq. (8))**	10%+W4A8	**3.49**	**6.61**	**5.32**	**5.71**	**9.25**	**63.49**	66.09	64.49	67.57	70.45
C	SVD-LLM (Wang et al., 2025)	80%	3.11	5.73	4.82	5.27	8.39	34.39	34.90	35.40	33.97	34.37
C	KFAC-SVD* (Thoma et al., 2026)	80%	3.11	5.73	4.82	5.27	8.39	34.87	35.12	35.62	35.27	34.57
Q	SpinQuant (Liu et al., 2025)	W3A8	3.04	5.61	4.81	5.20	8.27	61.25	**65.09**	**64.04**	65.62	69.13
Q	FlatQuant (Sun et al., 2025)	W3A8	3.04	5.61	4.81	5.20	8.27	**61.60**	64.58	62.59	**66.12**	68.88
Q	SERQ (Park et al., 2026)	W3A8	3.11	5.71	4.87	5.27	8.37	59.72	64.05	57.42	63.85	65.21
C+Q	**CoCaQ (Eq. (3))**	5%+W3A8	2.90	5.41	4.67	5.06	8.06	60.55	64.37	63.79	65.98	67.86
C+Q	**CoCaQ (Eq. (3))**	10%+W3A8	**2.77**	**5.17**	**4.54**	**4.93**	**7.76**	60.62	64.36	55.07	63.64	66.91
C+Q	**CoCaQ (Eq. (8))**	5%+W3A8	2.90	5.41	4.67	5.06	8.06	60.23	64.56	64.02	65.76	**69.73**
C+Q	**CoCaQ (Eq. (8))**	10%+W3A8	**2.77**	**5.17**	**4.54**	**4.93**	**7.76**	59.87	64.97	60.95	62.79	68.26
C	SVD-LLM (Wang et al., 2025)	85%	2.46	4.47	4.19	4.57	7.07	34.84	34.37	36.04	33.94	34.55
C	KFAC-SVD* (Thoma et al., 2026)	85%	2.46	4.47	4.19	4.57	7.07	34.73	34.82	35.10	34.90	34.16
Q	SpinQuant (Liu et al., 2025)	W2A8	2.24	4.02	3.95	4.33	6.61	43.13	46.54	41.97	52.78	57.67
Q	FlatQuant (Sun et al., 2025)	W2A8	2.24	4.02	3.95	4.33	6.61	35.18	34.49	35.42	34.36	35.50
Q	SERQ (Park et al., 2026)	W2A8	2.28	4.09	3.94	4.37	6.68	36.62	35.57	35.70	35.47	35.21
C+Q	**CoCaQ (Eq. (3))**	5%+W2A8	2.14	3.89	3.82	4.21	6.45	39.25	45.24	41.15	51.01	56.97
C+Q	**CoCaQ (Eq. (3))**	10%+W2A8	**2.05**	**3.73**	**3.76**	**4.15**	**6.27**	39.27	44.60	37.12	41.29	49.69
C+Q	**CoCaQ (Eq. (8))**	20%+W3A8	2.55	4.62	4.28	4.66	7.24	**53.02**	**58.63**	**42.35**	**54.30**	**58.99**
C+Q	**CoCaQ (Eq. (8))**	5%+W2A8	2.14	3.89	3.82	4.21	6.45	47.09	48.10	38.55	44.40	55.50
C+Q	**CoCaQ (Eq. (8))**	10%+W2A8	**2.05**	**3.73**	**3.76**	**4.15**	**6.27**	44.89	49.97	36.86	41.15	47.77

3.3 ASQ: ACTIVATION STABILIZATION FOR QUANTIZATION

Our QaMC improves the quality of the compressed model, but low-rank weight decomposition introduces an additional intermediate activation that is shifted from the original activation. Specifically, weight decomposition of matrix $\boldsymbol{W}$ as Eq. (8) produces an extra intermediate activation $\boldsymbol{H} = \widehat{\boldsymbol{B}}_r\boldsymbol{X}$, which typically exhibits enlarged dynamic ranges and severe outliers as shown in Fig. 2b, substantially increasing the difficulty of quantization. To alleviate this issue, we stabilize the activation by introducing an orthogonal regularizer $\boldsymbol{R}_*$ between the two low-rank factors as

$$\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{X} \approx \widehat{\boldsymbol{A}}_r\boldsymbol{R}_*\boldsymbol{R}_*^\top\widehat{\boldsymbol{B}}_r\boldsymbol{X} = \widehat{\boldsymbol{A}}_r\boldsymbol{R}_*\big(\boldsymbol{R}_*^\top\widehat{\boldsymbol{B}}_r\boldsymbol{X}\big) = \widehat{\boldsymbol{A}}_r\boldsymbol{R}_*\boldsymbol{H}', \tag{12}$$

where $\boldsymbol{R}_*\boldsymbol{R}_*^\top = \boldsymbol{I}$. Such orthogonal transformation aims to preserve the original linear mapping while reshaping the intermediate activation distributions by redistributing the activation energy across dimensions, thus mitigating compression-induced distribution shifts and reducing extreme outliers, as illustrated in Fig. 2b. This results in a narrower activation dynamic range and consequently reduces the quantization error, improving the robustness to post low-bit quantization, as shown in Appendix A.3 and Table 3. Since $\boldsymbol{R}_*$ can be folded into the weights, we incorporate it into each candidate low-rank layer during QLCA, ensuring that sensitivity is estimated on the actual compressed-and-stabilized structure. We construct the orthogonal matrices using QR decomposition throughout the work, while comparisons with Hadamard-based matrices are provided in Table 4.

4 EXPERIMENTS

We conduct experiments with LLaMA-2 (Touvron et al., 2023) 7B/13B, LLaMA-3 (Grattafiori et al., 2024) 8B, and Qwen3 (Yang et al., 2025) 8B/14B on WikiText-2 (Merity et al., 2016) for language modeling, MMLU (Hendrycks et al., 2020) for knowledge, and eight commonsense benchmarks for

*We reproduce KFAC-SVD+LEMS, which achieves the best performance in the original paper.

Table 2: Results of joint compression and quantization methods under W4A8 and W3A8 settings on LLaMA-2-7B and Qwen3-8B. For KFAC-SVD (Thoma et al., 2026) and SVD-LLM (Wang et al., 2025), we apply our quantization method after compression. We further adapt UniQL (Chiang et al., 2026) with our quantization scheme to ensure fair 8-bit activation quantization. All methods are evaluated without fine-tuning. Detailed results are provided in the Appendix A.11.

Setting		Method	LLaMA-2-7B			Qwen3-8B		
Quant.	Comp.		Wiki2↓	MMLU↑	CS Avg.↑	Wiki2↓	MMLU↑	CS Avg.↑
FP	–	Baseline	5.47	45.70	64.29	7.00	76.76	68.62
W4A8	5%	KFAC-SVD*	5.76	41.50	62.34	7.49	72.43	63.81
		SVD-LLM	8.47	25.69	49.09	10.63	47.36	58.71
		UniQL	5.89	**43.58**	62.63	8.12	72.25	68.35
		CoCaQ	**5.63**	43.31	**63.30**	**7.28**	**75.05**	**68.77**
	10%	KFAC-SVD*	5.93	38.12	62.46	7.76	70.96	59.26
		SVD-LLM	9.03	25.58	47.01	10.90	46.32	56.13
		UniQL	6.17	**42.59**	60.98	12.25	47.86	55.01
		CoCaQ	**5.71**	41.71	**63.49**	**7.45**	**72.69**	**67.57**
W3A8	5%	KFAC-SVD*	6.65	34.29	57.67	9.50	43.75	51.26
		SVD-LLM	14.33	25.62	40.54	16.09	28.53	48.46
		UniQL	7.61	26.30	57.57	10.05	57.63	60.62
		CoCaQ	**6.03**	**39.84**	**60.23**	**8.09**	**69.64**	**65.76**
	10%	KFAC-SVD*	7.28	29.49	55.35	10.69	36.69	49.26
		SVD-LLM	16.24	25.74	38.34	16.42	29.09	48.24
		UniQL	8.03	25.47	56.28	18.43	26.42	45.90
		CoCaQ	**6.15**	**36.78**	**59.87**	**8.50**	**66.92**	**62.79**
	15%	KFAC-SVD*	8.06	27.10	52.53	12.09	32.23	47.90
		SVD-LLM	17.39	25.57	38.38	21.33	26.11	43.28
		UniQL	9.03	24.30	53.65	19.44	27.76	45.74
		CoCaQ	**6.65**	**28.12**	**57.05**	**9.83**	**60.37**	**58.07**
	20%	KFAC-SVD*	8.79	**25.83**	49.84	13.49	32.69	45.68
		SVD-LLM	19.45	25.40	37.30	22.91	26.07	41.59
		UniQL	10.00	25.77	50.86	26.09	25.51	42.84
		CoCaQ	**7.87**	25.58	**53.02**	**13.01**	**45.95**	**54.30**

commonsense reasoning. For model compression, we randomly sample 128 samples from WikiText-2 for calibration. For quantization, the settings respectively follow those in GPTQ (Frantar et al., 2022) and SpinQuant (Liu et al., 2025) by using 128 WikiText-2 samples with a sequence length of 2048 for calibration and a group size of 128. We compare CoCaQ with several state-of-the-art model compression and quantization methods, and UniQL (Chiang et al., 2026) with joint of compression and quantization under the same settings. More details can be found in Appendix A.6.

4.1 MAIN RESULTS

First, we compare our CoCaQ with several state-of-the-art model compression and quantization methods under various memory budgets. As shown in Table 1, we have the following observations: (1) compression-only methods hardly work under low memory budgets. Specifically, the performance of SVD-LLM (Wang et al., 2025) and KFAC-SVD (Thoma et al., 2026) degrades considerably due to too high compression ratios under low-memory budgets; (2) Compared with quantization-only methods, CoCaQ achieves comparable or slightly better performance, while further reducing memory cost. For LLaMA-2-13B under W4A8, CoCaQ with 10% compression achieves 66.09 points with 6.61 GB of memory, closely matching SERQ (Park et al., 2026), which achieves 66.31 points but requires 7.33 GB of memory. They demonstrate that our CoCaQ can achieve a better trade-off between performance and memory than quantization-only methods; (3) CoCaQ can offer better flexibility, especially in the low-memory regime. For instance, CoCaQ with 20% compression and W3A8 achieves better performance than W2A8 quantization across all five models. They clearly verify the potential of effective coupling of low-rank weight decomposition and low-bit quantization in flexible yet efficient LLM deployment.

Furthermore, we compare with several joint compression and quantization methods with LLaMA-2-7B and Qwen3-8B under the settings of W4A8 and W3A8. As listed in Table 2, our CoCaQ achieves the best WikiText-2 perplexity and the highest CS Avg. among all models under the W3A8 setting, while consistently achieving the best MMLU accuracy on Qwen3-8B. Particularly, for Qwen3-8B, CoCaQ with 10% compression improves MMLU by 30.23 points and CS Avg. by 13.53 points over the second-best methods. Similarly, on LLaMA-2-7B, CoCaQ with 15% compression improves CS Avg. by 3.40 points and reduces PPL by 1.41. These results clearly demonstrate the robustness of our CoCaQ to challenging settings of low-bit quantization and high compression ratios.

4.2 Ablation Study

Component Ablation. Table 3 presents the ablation results of CoCaQ on LLaMA-2-7B under 5%+W3A8. Both QaMC and ASQ consistently improve performance, while their combination achieves the best results across all metrics, demonstrating their complementary effects.

QF-SVD Selection. Fig. 4 compares the QF-SVD-W (Eq. (3)) and QF-SVD-A (Eq. (8)), under W3A8 across different compression ratios. QF-SVD-A (Eq. (8)) generally achieves better performance. In contrast, aggressive compression increases the reconstruction error of QF-SVD-W (Eq. (3)), which limits its performance. Detailed analysis is provided in Appendix A.4.

Table 3: Effect of key component.

Setting	QaMC	ASQ	Performance		
			Wiki2 ↓	MMLU ↑	CS Avg. ↑
Baseline			6.48	31.11	58.14
	✓		6.14	35.61	59.26
		✓	6.25	38.87	58.77
Full method	✓	✓	6.03	39.84	60.23

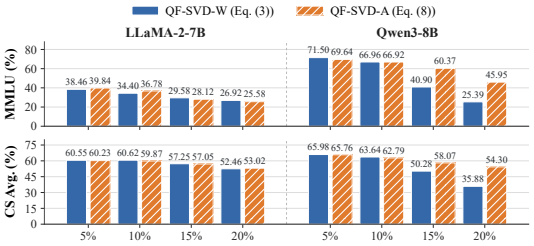


Figure 4: Comparison of QF-SVD strategies.

$\boldsymbol{R}_*$ Initialization. Table 4 compares Hadamard- and QR-based random orthogonal initialization for $\boldsymbol{R}_*$ under 5%+W3A8. QR initialization achieves better overall performance. The training of $\boldsymbol{R}_*$ following SpinQuant (Liu et al., 2025) brings few gains but more computational cost. Stability verification of the two initialization schemes is provided in the Appendix A.13.

Table 4: Comparison of various $\boldsymbol{R}_*$.

Method	Wiki2↓	MMLU↑	CS Avg.↑
QR	6.03	39.84	60.23
+train	6.03	39.60	60.76
Hadamard	6.02	38.06	60.53
+train	6.02	39.31	60.55

Table 5: Compression-Quantization allocation.

Setting	Memory	Wiki2↓	MMLU↑	CS Avg.↑
30%+W4A8	2.83 GB	10.01	25.17	49.01
10%+W3A8	2.77 GB	6.15	36.78	59.87
25%+W3A8	2.40 GB	10.20	24.73	48.39
5%+W2A8	2.14 GB	15.98	25.83	47.09

Compression-Quantization Allocation. Table 5 shows that, under similar memory budgets, lower compression with lower-bit quantization performs better in the W4–W3 regime, whereas higher compression with higher-bit quantization performs better in the W3–W2 regime. They highlight flexibility of CoCaQ in unifying compression and quantization for accuracy–memory trade-offs.

5 Conclusion

In this paper, we presented CoCaQ, a training-free method to couple compression and quantization by accounting for their mutual influence during the joint working. Through QaMC and ASQ, CoCaQ constructs quantization-friendly compressed models while mitigating compression-induced activation shifts. Experiments on LLaMA and Qwen demonstrate favorable memory-accuracy trade-offs, particularly under low memory budgets. In future work, we will extend CoCaQ to multimodal large language models and explore the joint compression and quantization of both language and visual components for flexible and efficient deployment.

AI USE STATEMENT

In this work, we used generative AI tools for manuscript language polishing and LaTeX formatting. We have not used generative AI tools for experimental design, data collection, data analysis, model training, or result interpretation. We have reviewed all AI-assisted work by manually checking the revised text and LaTeX structure for accuracy and consistency. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, personally identifiable information, or sensitive data. We are not aware of any ethical concerns associated with the methodology or experiments presented in this work.

REPRODUCIBILITY STATEMENT

We provide detailed information to facilitate the reproducibility of CoCaQ. The implementation details of the proposed QaMC and ASQ are described in Section 3, including the corresponding algorithms. The experimental setup, including model configurations, calibration data, memory budgets, compression ratios, quantization settings, and evaluation protocols, is provided in Section 4. Additional implementation details, ablation studies, and experimental results are included in the Appendix A.

REFERENCES

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Mohamed S Abdelfattah, and Kai-Chiang Wu. Palu: Compressing kv-cache with low-rank projection. *arXiv preprint arXiv:2407.21118*, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Hung-Yueh Chiang, Chi-Chih Chang, Yu-Chen Lu, Chien-Yu Lin, Kai-Chiang Wu, Mohamed Abdelfattah, and Diana Marculescu. Uniql: Unified quantization and low-rank compression for adaptive edge llms. In *International Conference on Learning Representations*, volume 2026, pp. 32286–32310, 2026.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pp. 2924–2936, 2019.

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

Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. *Advances in neural information processing systems*, 27, 2014.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in neural information processing systems*, 35: 30318–30332, 2022.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

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

Prakhar Ganesh, Yao Chen, Xin Lou, Mohammad Ali Khan, Yin Yang, Hassan Sajjad, Preslav Nakov, Deming Chen, and Marianne Winslett. Compressing large-scale transformer-based models: A case study on bert. *Transactions of the Association for Computational Linguistics*, 9: 1061–1080, 2021.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models. *arXiv preprint arXiv:2411.05007*, 2024.

Yesheng Liang, Haisheng Chen, Song Han, and Zhijian Liu. Paroquant: Pairwise rotation quantization for efficient reasoning llm inference. In *International Conference on Learning Representations*, volume 2026, pp. 77000–77017, 2026.

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

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations. In *International Conference on Learning Representations*, volume 2025, pp. 92009–92032, 2025.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv:2203.13474*, 2022.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Yeonsik Park, Hyeonseong Kim, and Seungkyu Choi. SERQ: Saliency-aware low-rank error reconstruction for LLM quantization. In *International Conference on Learning Representations*, 2026.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 4463–4473, 2019.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, et al. Flatquant: Flatness matters for llm quantization. In *International conference on machine learning*, 2025.

Moritz Thoma, Maximilian Groezinger, Maximilian Forstenhäusler, Emad Aghajanzadeh, Manoj Rohit Vemparala, Christos Anagnostopoulos, Pierpaolo Mori, Nael Fasfous, Alexander Frickenstein, Daniel Mueller-Gritschneder, et al. Advancing svd-based llm compression via layer-wise error model search. In *Forty-third International Conference on Machine Learning*, 2026.

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

Haiyu Wang, Yutong Wang, Jack Jiang, and Sai Qian Zhang. Wsvd: Weighted low-rank approximation for fast and efficient execution of low-precision vision-language models. *arXiv preprint arXiv:2604.02570*, 2026a.

Haiyu Wang, Yutong Wang, Leshu Li, Yihui Ren, and Sai Qian Zhang. Laser: Loss-aware singular-value decomposition and rank allocation for efficient low-precision vision-language models. *arXiv preprint arXiv:2606.00573*, 2026b.

Xin Wang, Samiul Alam, Zhongwei Wan, Hui Shen, and Mi Zhang. Svd-llm v2: Optimizing singular value truncation for large language model compression. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4287–4296, 2025.

Yutong Wang, Haiyu Wang, and Sai Qian Zhang. Qsvd: Efficient low-rank approximation for unified query-key-value weight compression in low-precision vision-language models. *Advances in Neural Information Processing Systems*, 38:1789–1820, 2026c.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Jingyang Xiang and Sai Qian Zhang. Dfrot: Achieving outlier-free and massive activation-free for rotated llms with refined rotation. *arXiv preprint arXiv:2412.00648*, 2024.

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

Lin Xv, Xian Gao, Ting Li, and Yuzhuo Fu. Beyond uniform svd: Dual-level optimization across columns and modules for llm compression. *arXiv preprint arXiv:2510.19385*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Hao Yu and Jianxin Wu. Compressing transformers: Features are low-rank, but weights are not! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11007–11015, 2023.

Zhihang Yuan, Yuzhang Shang, Yue Song, Dawei Yang, Qiang Wu, Yan Yan, and Guangyu Sun. Asvd: Activation-aware singular value decomposition for compressing large language models. *arXiv preprint arXiv:2312.05821*, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

A Appendix

A.1 QF-SVD Reconstruction Error Analysis

Let $\widetilde{\boldsymbol{W}} = \boldsymbol{R}_1 \boldsymbol{W} \boldsymbol{R}_2$ and $\widetilde{\boldsymbol{X}} = \boldsymbol{R}_2^\top \boldsymbol{X}$, where $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are orthogonal matrices. Given $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, we have

$$\widetilde{\boldsymbol{W}} = (\boldsymbol{R}_1\boldsymbol{U})\boldsymbol{\Sigma}(\boldsymbol{R}_2^\top \boldsymbol{V})^\top.$$

Given the SVD of $\boldsymbol{W}$,

$$\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top,$$

the orthogonally reparameterized weight matrix can be written as

$$\begin{aligned}\widetilde{\boldsymbol{W}} &= \boldsymbol{R}_1 \boldsymbol{W} \boldsymbol{R}_2 \\ &= \boldsymbol{R}_1 \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \boldsymbol{R}_2 \\ &= (\boldsymbol{R}_1\boldsymbol{U})\boldsymbol{\Sigma}(\boldsymbol{R}_2^\top \boldsymbol{V})^\top.\end{aligned}$$

Since $\boldsymbol{R}_1$, $\boldsymbol{R}_2$, $\boldsymbol{U}$, and $\boldsymbol{V}$ are orthogonal, the transformed singular-vector matrices remain orthogonal:

$$(\boldsymbol{R}_1\boldsymbol{U})^\top(\boldsymbol{R}_1\boldsymbol{U}) = \boldsymbol{U}^\top \boldsymbol{R}_1^\top \boldsymbol{R}_1 \boldsymbol{U} = \boldsymbol{I},$$

and

$$(\boldsymbol{R}_2^\top\boldsymbol{V})^\top(\boldsymbol{R}_2^\top\boldsymbol{V}) = \boldsymbol{V}^\top \boldsymbol{R}_2 \boldsymbol{R}_2^\top \boldsymbol{V} = \boldsymbol{I}.$$

Therefore, $(\boldsymbol{R}_1\boldsymbol{U})\boldsymbol{\Sigma}(\boldsymbol{R}_2^\top\boldsymbol{V})^\top$ satisfies the defining conditions of an SVD of $\widetilde{\boldsymbol{W}}$. Importantly, the diagonal matrix $\boldsymbol{\Sigma}$ is unchanged, indicating that orthogonal left- and right-multiplication only rotates the singular vectors without changing the singular values. Hence,

$$\sigma_i(\widetilde{\boldsymbol{W}}) = \sigma_i(\boldsymbol{W}), \quad \forall i.$$

Although the singular vectors returned by a numerical SVD may differ due to sign or basis ambiguities, the singular values are invariant under orthogonal reparameterization. For the rank-r approximation $\boldsymbol{W}_r = \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top$, the corresponding approximation after orthogonal reparameterization is

$$\widetilde{\boldsymbol{W}}_r = \boldsymbol{R}_1 \boldsymbol{W}_r \boldsymbol{R}_2.$$

Together with

$$\widetilde{\boldsymbol{W}} = \boldsymbol{R}_1 \boldsymbol{W} \boldsymbol{R}_2, \qquad \widetilde{\boldsymbol{X}} = \boldsymbol{R}_2^\top \boldsymbol{X},$$

the reconstruction error becomes

$$\begin{aligned}\widetilde{\mathcal{L}}_{\text{rec}} &= \left\| \widetilde{\boldsymbol{W}}\widetilde{\boldsymbol{X}} - \widetilde{\boldsymbol{W}}_r\widetilde{\boldsymbol{X}} \right\|_{\text{F}} \\ &= \left\| \boldsymbol{R}_1\boldsymbol{W}\boldsymbol{R}_2\boldsymbol{R}_2^\top\boldsymbol{X} - \boldsymbol{R}_1\boldsymbol{W}_r\boldsymbol{R}_2\boldsymbol{R}_2^\top\boldsymbol{X} \right\|_{\text{F}} \\ &= \left\| \boldsymbol{R}_1\boldsymbol{W}\boldsymbol{X} - \boldsymbol{R}_1\boldsymbol{W}_r\boldsymbol{X} \right\|_{\text{F}} \\ &= \left\| \boldsymbol{R}_1(\boldsymbol{W} - \boldsymbol{W}_r)\boldsymbol{X} \right\|_{\text{F}}.\end{aligned}$$

Since $\boldsymbol{R}_1$ is orthogonal, the Frobenius norm is invariant under left multiplication by $\boldsymbol{R}_1$. Specifically, for any matrix $\boldsymbol{A}$,

$$\begin{aligned}\|\boldsymbol{R}_1\boldsymbol{A}\|_{\text{F}}^2 &= \operatorname{tr}\left((\boldsymbol{R}_1\boldsymbol{A})^\top(\boldsymbol{R}_1\boldsymbol{A})\right) \\ &= \operatorname{tr}\left(\boldsymbol{A}^\top\boldsymbol{R}_1^\top\boldsymbol{R}_1\boldsymbol{A}\right) \\ &= \operatorname{tr}\left(\boldsymbol{A}^\top\boldsymbol{A}\right) \\ &= \|\boldsymbol{A}\|_{\text{F}}^2.\end{aligned}$$

Therefore,

$$\begin{aligned}\widetilde{\mathcal{L}}_{\text{rec}} &= \|\boldsymbol{R}_1(\boldsymbol{W} - \boldsymbol{W}_r)\boldsymbol{X}\|_{\text{F}} \\ &= \|(\boldsymbol{W} - \boldsymbol{W}_r)\boldsymbol{X}\|_{\text{F}} \\ &= \|\boldsymbol{W}\boldsymbol{X} - \boldsymbol{W}_r\boldsymbol{X}\|_{\text{F}} \\ &= \mathcal{L}_{\text{rec}}.\end{aligned}$$

Thus, when the input activations are transformed consistently, orthogonal reparameterization before SVD preserves the reconstruction error of the rank-r approximation.

A.2 QF-SVD Quantization Error Analysis

To investigate the quantization friendliness of QF-SVD, we analyze the weight distributions as well as the corresponding quantization errors. We first consider the effect of the orthogonal reparameterization in QF-SVD Eq. (3). Specifically, we compare the weight distributions of the proposed method with those of standard SVD for the same layer under 10% compression. As shown in Fig. 5(a) and (b), orthogonal reparameterization substantially reduces the magnitude of outliers in weights, leading to a more compact and smoother distribution. This improved representation better preserves information under low-bit quantization.

To quantitatively evaluate its impact on weight quantization, we consider the W3A16 setting and compute the quantization error

$$\mathcal{L}_{\text{quant}} = \|\boldsymbol{A}_r\boldsymbol{B}_r\boldsymbol{X} - \mathrm{Q}(\boldsymbol{A}_r)\,\mathrm{Q}(\boldsymbol{B}_r)\boldsymbol{X}\|_{\mathrm{F}}\,.$$

For a clearer comparison, we normalize $\mathcal{L}_{\text{quant}}$ by that of standard SVD. As shown in Fig. 5(c), QF-SVD Eq. (3) substantially reduces $\mathcal{L}_{\text{quant}}$, demonstrating that the orthogonal reparameterization produces a more quantization-friendly low-rank representation.

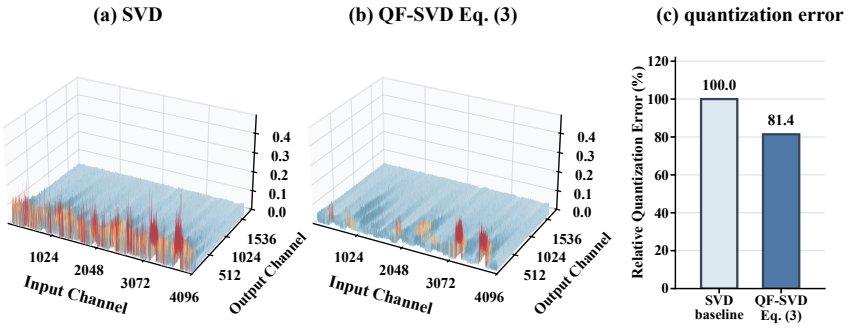


Figure 5: Visualization of the effect of QF-SVD(Eq. (3))

Energy Redistributing. After decomposing the weight matrix using Eq. (5), we observe that the two low-rank factors exhibit substantially imbalanced energy distributions. In particular, the weights in $\boldsymbol{B}_r$ contain more pronounced outliers, whereas those in $\boldsymbol{A}_r$ are relatively smoother, indicating that the two factors have different quantization difficulties, as shown in Fig. 6. To alleviate this imbalance, we redistribute the energy between the two factors and obtain $\widehat{\boldsymbol{A}}_r$ and $\widehat{\boldsymbol{B}}_r$ in Eq. (8). This redistribution preserves their product while equalizing their Frobenius-norm energy, i.e., $|\widehat{\boldsymbol{A}}_r|_F^2 = |\widehat{\boldsymbol{B}}_r|_F^2 = \mathrm{tr}(\boldsymbol{\Sigma}_b)$. The proof is given below.

Based on that $\boldsymbol{Q}_r^\top\boldsymbol{Q}_r = \boldsymbol{I}$, $\boldsymbol{U}_b^\top\boldsymbol{U}_b = \boldsymbol{I}$, and $\boldsymbol{V}_b^\top\boldsymbol{V}_b = \boldsymbol{I}$, we have

$$\begin{aligned} ||\widehat{\boldsymbol{A}}_r||_F^2 &= \mathrm{tr}\left(\widehat{\boldsymbol{A}}_r^\top\widehat{\boldsymbol{A}}_r\right) \\ &= \mathrm{tr}\left(\boldsymbol{\Sigma}_b^{1/2}\boldsymbol{U}_b^\top\boldsymbol{Q}_r^\top\boldsymbol{Q}_r\boldsymbol{U}_b\boldsymbol{\Sigma}_b^{1/2}\right) \\ &= \mathrm{tr}\left(\boldsymbol{\Sigma}_b^{1/2}\boldsymbol{U}_b^\top\boldsymbol{U}_b\boldsymbol{\Sigma}_b^{1/2}\right) \\ &= \mathrm{tr}\left(\boldsymbol{\Sigma}_b\right). \end{aligned} \tag{13}$$

Similarly,

$$\begin{aligned} ||\widehat{\boldsymbol{B}}_r||_F^2 &= \mathrm{tr}\left(\widehat{\boldsymbol{B}}_r\widehat{\boldsymbol{B}}_r^\top\right) \\ &= \mathrm{tr}\left(\boldsymbol{\Sigma}_b^{1/2}\boldsymbol{V}_b^\top\boldsymbol{V}_b\boldsymbol{\Sigma}_b^{1/2}\right) \\ &= \mathrm{tr}\left(\boldsymbol{\Sigma}_b\right). \end{aligned} \tag{14}$$

Therefore,

$$||\widehat{\boldsymbol{A}_r}||_F^2 = ||\widehat{\boldsymbol{B}_r}||_F^2 = \mathrm{tr}(\boldsymbol{\Sigma}_b). \tag{15}$$

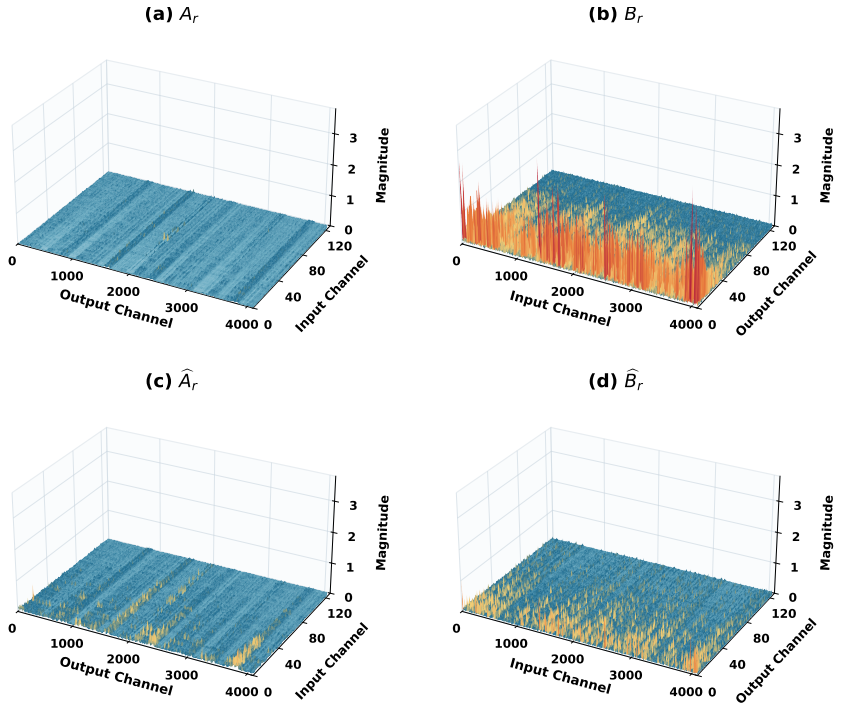


Figure 6: Visualization of imbalanced energy distributions in QF-SVD(Eq. (5))

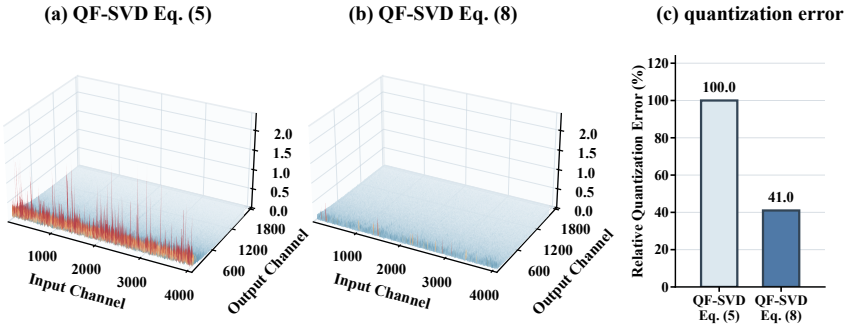


Figure 7: Visualization of the effect of QF-SVD (Eq. (8)) under weight quantization.

We also investigate the effect of redistributing the energy between the low-rank factors in Eq. (8). Following the same evaluation protocol, we compare the weight distributions before and after energy redistribution. We use the quantization error of the compressed model before energy redistribution, corresponding to Eq. (5), as the normalization baseline. As shown in Fig. 7, energy redistribution makes the weight distributions more balanced and substantially suppresses outliers, resulting in a reduction of $\mathcal{L}_{\text{quant}}$ by up to 59%. This further demonstrates the quantization-friendly property of QF-SVD Eq. (8).

Finally, we examine the impact of energy redistribution on activation quantization under the W16A8 setting. As shown in Fig. 8, redistributing the factor energy also produces a smoother activation distribution with fewer outliers. Correspondingly, the activation quantization error is reduced by 27.1%, indicating that energy redistribution improves not only weight quantization but also the robustness of activations to low-bit quantization.

A.3 ASQ Analysis

Following the analysis of $\mathcal{L}_{\text{quant}}$ in Sec. A.2, we compare three variants: QF-SVD without energy redistribution (Eq. (5)), QF-SVD with energy redistribution (Eq. (8)), and QF-SVD with energy

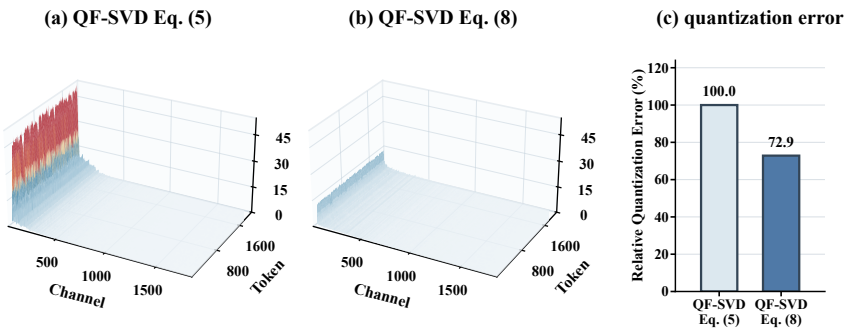


Figure 8: Visualization of the effect of QF-SVD (Eq. (8)) under activation quantization.

redistribution followed by ASQ. We evaluate these variants under W3A16, W16A8, and W3A8 settings, as shown in Fig. 9(a), (b), and (c), respectively. The results show that incorporating ASQ consistently further reduces $\mathcal{L}_{\text{quant}}$ across weight-only, activation-only, and joint weight-activation quantization, demonstrating its effectiveness in mitigating the quantization error induced by the compressed representation.

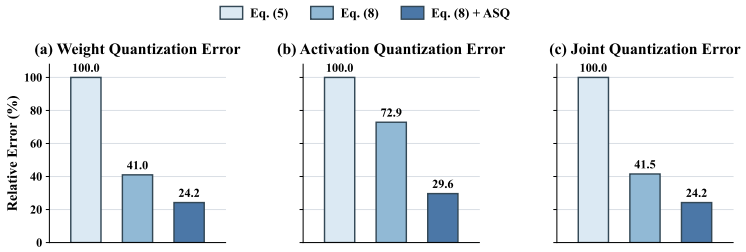


Figure 9: Quantization error analysis of ASQ

A.4 Reconstruction Error Comparison of Two QF-SVD

As shown in Fig. 10, the reconstruction error of QF-SVD in Eq. 3 grows more rapidly as the compression ratio increases. In contrast, QF-SVD in Eq. 8 substantially reduces the reconstruction error, demonstrating greater robustness to aggressive compression.

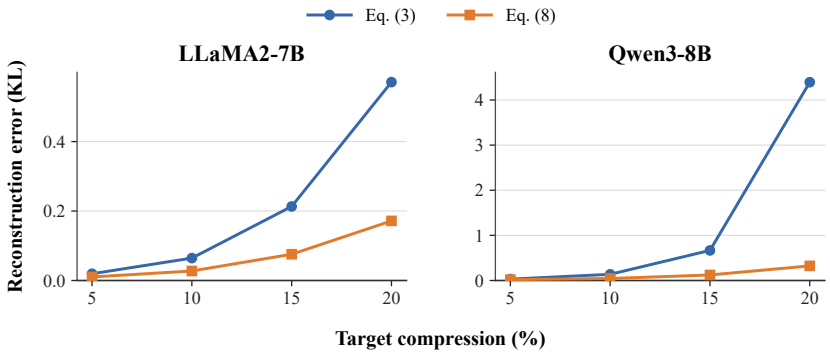


Figure 10: Reconstruction Error Comparison of Two QF-SVD.

A.5 Rank-Cutoff Binary Search

The detailed binary search procedure for QLCA is presented in Algorithm 2. The candidate ranks, parameter savings, and KL sensitivities used here are computed by the candidate-evaluation stage of CoCaQ before the binary search. Each layer is also assigned a keep-original candidate, while

duplicate or dominated candidates are removed. The remaining compressed candidates are merged into $\mathbb{C}$ and sorted from larger to smaller sensitivity. The two pointers initially cover indices 0 to $N-1$. At each iteration, their midpoint m defines the suffix $\mathbb{C}[m : N-1]$. For each layer, the algorithm selects from this suffix the candidate rank with the largest parameter saving; if no candidate for that layer remains, it retains the original rank. It then sums the selected candidates to obtain the total parameter saving ΔP_m and total sensitivity $\mathcal{L}_m$.

The search compares ΔP_m with the target saving T implied by budget B. If $\Delta P_m \geq T$, the trial structure meets the target, so the low pointer moves to m to test a stricter cutoff that removes more high-damage candidates. If $\Delta P_m < T$, the high pointer moves to m because the trial structure saves too few parameters. Increasing m cannot increase the achievable saving, so the interval contracts until the two pointers are adjacent. Finally, the algorithm computes $(\Delta P_{low}, \mathcal{L}_{low})$ and $(\Delta P_{high}, \mathcal{L}_{high})$ for the two endpoint rank selections. The selection closer to the target T is denoted by $\{r_l^*\}_{l=1}^L$; if both gaps are equal, the one with smaller total KL sensitivity is used.

Algorithm 2 Rank-Cutoff Binary Search

Input: For each layer l, candidate ranks $\mathbb{R}_l$ with parameter savings $\Delta P_l(r)$ and sensitivities $\mathcal{L}_{\mathrm{Q}}(l, r)$; target saving T implied by budget B
Output: Selected ranks $\{r_l^*\}_{l=1}^L$ closest to the target T
Merge all compressed candidates into $\mathbb{C}$ and sort by decreasing $\mathcal{L}_{\mathrm{Q}}(l, r)$
Let N denote the number of compressed candidates: $N \leftarrow |\mathbb{C}|$
$low \leftarrow 0$, $high \leftarrow N-1$
while $high - low > 1$ **do**
    $m \leftarrow \lfloor (low + high)/2 \rfloor$
    for all layers l **do**
        if $\mathbb{C}[m : N-1]$ contains a candidate from $\mathbb{R}_l$ **then**
            $r_l^{(m)} \leftarrow$ the candidate rank with largest $\Delta P_l(r)$
        else
            $r_l^{(m)} \leftarrow$ the original rank of layer l
        End if
    End For
    $\Delta P_m \leftarrow \sum_l \Delta P_l(r_l^{(m)})$, $\mathcal{L}_m \leftarrow \sum_l \mathcal{L}_{\mathrm{Q}}(l, r_l^{(m)})$
    if $\Delta P_m \geq T$ **then**
        $low \leftarrow m$
    else
        $high \leftarrow m$
    End if
End while
Compute $(\Delta P_{low}, \mathcal{L}_{low})$ for $\{r_l^{(low)}\}_{l=1}^L$ as above
Compute $(\Delta P_{high}, \mathcal{L}_{high})$ for $\{r_l^{(high)}\}_{l=1}^L$ as above
$\{r_l^*\}_{l=1}^L \leftarrow \{r_l^{(low)}\}_{l=1}^L$ or $\{r_l^{(high)}\}_{l=1}^L$ according to the smaller of $|\Delta P_{low} - T|$ and $|\Delta P_{high} - T|$
if $|\Delta P_{low} - T| = |\Delta P_{high} - T|$ **then**
    $\{r_l^*\}_{l=1}^L \leftarrow$ the one with smaller $\mathcal{L}_{low}$ or $\mathcal{L}_{high}$
End if
Return $\{r_l^*\}_{l=1}^L$

A.6 Detailed experiments

Models and Tasks. We conduct experiments on the LLaMA-2 (Touvron et al., 2023) 7B/13B, LLaMA-3 (Grattafiori et al., 2024) 8B, and Qwen3 (Yang et al., 2025) 8B/14B models. We evaluate CoCaQ on three benchmark categories: (1) *Language Modeling:* Perplexity on WikiText-2 (Merity et al., 2016). (2) *Knowledge Understanding:* 5-shot accuracy on MMLU (Hendrycks et al., 2020). (3) *Commonsense Reasoning:* Zero-shot accuracy on BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC-Easy, ARC-Challenge (Clark et al., 2018), and OpenBookQA (Mihaylov et al., 2018).

Implementation. *For compression*, activation-based decomposition is adopted across all compression ratios unless otherwise specified. For all methods that require dataset for compression, we randomly sample 128 sequences from WikiText-2 (Merity et al., 2016) for calibration. *For weight*

quantization, we adopt GPTQ (Frantar et al., 2022) as the post-training quantization framework. Following the standard GPTQ configuration, we use 128 samples from WikiText-2 with a sequence length of 2048 as the calibration set and set the group size to 128. *For activation quantization*, following (Liu et al., 2025), we apply an orthogonal rotation before the down layer and use round-to-nearest(RTN) for activation quantization.

Baselines. We compare CoCaQ with three categories of representative baselines. (1) *Compression*: We select several state-of-the-art low-rank decomposition methods, including SVD-LLM (Wang et al., 2025) and KFAC-SVD (Thoma et al., 2026). (2) *Quantization*: We compare against representative quantization approaches, including SpinQuant (Liu et al., 2025), FlatQuant (Sun et al., 2025) and SERQ (Park et al., 2026). (3) *Compression-quantization combination*: We include UniQL (Chiang et al., 2026), a unified compression and quantization framework. For fair comparison, all baseline results are reproduced using their official codebases under the same experimental setting.

A.7 Speed Measurement

We conduct an end-to-end speed measurement of the LLaMA-2 7B model under W16A16 and W4A8 configurations on Intel(R) Xeon(R) Platinum 8369B CPU. As shown in Table 6, additional compression beyond quantization reduces the overall computational workload and decreases the inference time.

Table 6: Real-time end-to-end speed measurement of LLaMA-2 7B.

Method	Configuration	Decoding speed(ms/token)	Computation(GMACs/token)
Baseline	W16A16	5301.07	6.61
SpinQuant	W4A8	169.85	6.61
CoCaQ	W4A8+5%	166.87	6.28
	W4A8+10%	164.70	5.96
	W4A8+20%	158.71	5.31

A.8 Flexible Compression–Quantization Trade-offs

Figure 1 illustrates how different compression and quantization strategies trade accuracy for memory consumption. Compression-only and quantization-only methods provide a small number of discrete operating points, whereas coupling methods allow the compression ratio and weight precision to be adjusted independently. Table 7 reports the exact values used in the figure.

Table 7: Memory (GB) and CS Avg. in Figure 1.

Type	Method	Setting	LLaMA-2-7B		LLaMA-2-13B	
			Memory (GB) ↓	CS Avg. ↑	Memory (GB) ↓	CS Avg. ↑
-	Baseline	FP	13.50	64.29	26.00	66.81
C	SVD-LLM	75%	3.75	34.21	7.00	34.70
		80%	3.11	34.39	5.73	34.90
		85%	2.46	34.84	4.47	34.37
	KFAC-SVD	75%	3.75	35.33	7.00	35.12
		80%	3.11	34.87	5.73	35.12
		85%	2.46	34.73	4.47	34.82
Q	SpinQuant	W4A8	3.86	63.11	7.19	66.11
		W3A8	3.04	61.25	5.61	65.09
		W2A8	2.24	43.13	4.02	46.54
	SERQ	W4A8	3.95	63.34	7.33	66.31
		W3A8	3.11	59.72	5.71	64.05
		W2A8	2.28	36.62	4.09	35.57
C+Q	UniQL	5% + W4A8	3.66	62.63	6.84	65.77
		5% + W3A8	2.90	57.57	5.41	62.27
		10% + W3A8	2.77	56.28	5.17	62.39
		15% + W3A8	2.65	53.65	4.93	60.75
		20% + W3A8	2.55	50.86	4.62	58.35
		5% + W2A8	2.14	34.87	3.89	35.69
	CoCaQ (Eq.(8))	5% + W4A8	3.66	63.30	6.84	66.07
		5% + W3A8	2.90	60.23	5.41	64.56
		10% + W3A8	2.77	59.87	5.17	64.97
		15% + W3A8	2.65	57.05	4.93	63.46
		20% + W3A8	2.55	53.02	4.62	58.63
		5% + W2A8	2.14	47.09	3.89	48.10

A.9 Quantization Behavior of Different Compressed Structures

Figure 2a shows that different compressed structures exhibit similar performance before quantization but substantially different performance after quantization. Table 8 reports the corresponding results under the 5%+W3A8 setting. Figure 11 shows the heatmap of different compressed structures.

Table 8: Effect of quantization on compressed structures.

Method	Before quantization			After quantization		
	Wiki2 ↓	MMLU ↑	CS Avg. ↑	Wiki2 ↓	MMLU ↑	CS Avg. ↑
ASVD	5.62	40.93	63.22	6.50	36.59	59.87
LEMS	5.72	44.46	63.92	6.42	35.66	60.43
QLCA	5.57	43.83	63.92	6.27	38.46	60.55

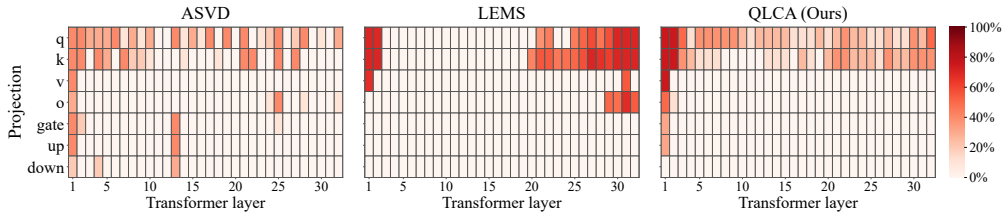


Figure 11: Layer-wise parameter reduction ratios under different compression settings.

A.10 Performance Comparison Across Models

The following tables report model-wise quantization results across the evaluated LLaMA and Qwen models.

Table 9: Evaluation results for LLaMA-2-7B across quantization methods and settings.

Type	Method	Setting	Memory	Arc_e	Arc_c	Openb.	BoolQ	PIQA	HellaS.	WinoG.	SIQA	CS Avg
-	Baseline	FP	13.50	76.26	46.08	44.00	77.92	78.94	75.93	69.22	45.96	64.29
C	SVD-LLM	75%	3.75	25.76	26.54	25.80	37.83	50.44	26.15	46.80	34.34	34.21
	KFAC-SVD*	75%	3.75	27.31	26.11	26.60	37.83	50.92	27.57	51.07	35.26	35.33
Q	SpinQuant	W4A8	3.86	75.08	44.54	42.60	75.38	78.13	75.19	68.43	45.50	63.11
	SERQ	W4A8	3.95	75.13	44.71	43.20	75.23	78.94	75.16	69.30	45.04	63.34
	FlatQuant	W4A8	3.86	76.05	44.11	43.00	76.09	78.62	75.00	68.43	45.96	63.41
C+Q	**CoCaQ (Eq.(3))**	5%+W4A8	3.66	76.05	44.37	44.00	74.77	78.84	74.75	68.67	45.65	63.39
	CoCaQ (Eq.(3))	10%+W4A8	3.49	74.07	42.58	43.80	75.44	77.42	73.89	68.43	45.09	62.59
	CoCaQ (Eq.(8))	5%+W4A8	3.66	76.01	43.69	44.00	76.12	78.35	74.21	68.51	45.55	63.30
	CoCaQ (Eq.(8))	10%+W4A8	3.49	74.87	43.34	44.20	77.19	79.27	74.45	69.77	44.83	63.49
C	SVD-LLM	80%	3.11	25.88	27.82	23.40	37.83	49.89	25.79	50.67	33.83	34.39
	KFAC-SVD*	80%	3.11	27.19	27.22	26.60	37.83	50.16	26.39	48.54	35.06	34.87
Q	SpinQuant	W3A8	3.04	72.90	41.89	40.40	73.27	77.69	72.08	68.11	43.65	61.25
	SERQ	W3A8	3.11	69.28	38.65	41.80	70.37	75.95	70.63	67.09	44.01	59.72
	FlatQuant	W3A8	3.04	72.01	40.87	41.00	77.28	76.99	72.20	67.40	45.09	61.60
C+Q	**CoCaQ (Eq.(3))**	5%+W3A8	2.90	73.02	39.59	39.20	74.01	76.61	71.12	67.56	43.30	60.55
	CoCaQ (Eq.(3))	10%+W3A8	2.77	72.43	40.02	43.20	72.84	76.44	69.63	67.48	42.94	60.62
	CoCaQ (Eq.(8))	5%+W3A8	2.90	72.31	40.02	40.80	70.98	77.20	71.13	66.54	42.84	60.23
	CoCaQ (Eq.(8))	10%+W3A8	2.77	71.46	39.42	40.80	71.56	76.88	69.96	65.82	43.04	59.87
C	SVD-LLM	85%	2.46	26.73	28.07	25.00	37.80	49.84	26.00	50.91	34.34	34.84
	KFAC-SVD*	85%	2.46	26.18	26.28	26.00	37.83	50.33	25.89	50.51	34.85	34.73
Q	SpinQuant	W2A8	2.24	43.86	24.49	28.40	61.16	58.92	37.05	55.09	36.08	43.13
	SERQ	W2A8	2.28	25.25	28.75	27.60	50.21	51.41	26.01	49.64	34.08	36.62
	FlatQuant	W2A8	2.24	26.85	27.47	23.80	40.06	54.03	25.33	49.09	34.80	35.18
C+Q	**CoCaQ (Eq.(3))**	5%+W2A8	2.14	33.04	21.42	26.60	61.10	54.79	30.16	51.07	35.82	39.25
	CoCaQ (Eq.(3))	10%+W2A8	2.05	33.25	22.87	25.40	61.83	56.47	29.32	50.12	34.90	39.27
	CoCaQ (Eq.(8))	20%+W3A8	2.55	63.51	32.00	37.00	63.94	71.33	57.56	60.93	37.93	53.02
	CoCaQ (Eq.(8))	5%+W2A8	2.14	53.45	26.79	31.80	61.50	63.66	46.04	57.06	36.44	47.09
	CoCaQ (Eq.(8))	10%+W2A8	2.05	46.17	23.89	31.80	62.45	60.77	41.43	56.43	36.18	44.89

Table 10: Evaluation results for LLaMA-2-13B across quantization methods and settings.

Type	Method	Setting	Memory	Arc_e	Arc_c	Openb.	BoolQ	PIQA	HellaS.	WinoG.	SIQA	CS Avg
-	Baseline	FP	26.00	79.63	49.40	45.60	80.67	80.47	79.38	71.74	47.60	66.81
C	SVD-LLM	75%	7.00	26.52	27.05	25.60	37.83	49.40	26.24	50.04	34.90	34.70
	KFAC-SVD*	75%	7.00	28.62	25.09	25.00	37.83	51.20	28.27	50.28	34.65	35.12
Q	SpinQuant	W4A8	7.19	78.41	48.46	45.00	79.91	80.09	78.62	71.98	46.42	66.11
	SERQ	W4A8	7.33	79.21	48.89	45.40	77.61	80.30	78.88	73.48	46.72	66.31
	FlatQuant	W4A8	7.19	77.74	47.95	44.60	79.72	80.36	78.70	72.53	46.47	66.01
C+Q	**CoCaQ (Eq.(3))**	5%+W4A8	6.84	78.45	47.87	45.60	78.65	79.92	78.61	72.14	46.62	65.98
	CoCaQ (Eq.(3))	10%+W4A8	6.61	78.32	48.46	44.20	79.17	79.65	78.93	71.82	45.75	65.79
	CoCaQ (Eq.(8))	5%+W4A8	6.84	78.54	47.44	45.80	80.73	79.98	78.42	71.74	45.91	66.07
	CoCaQ (Eq.(8))	10%+W4A8	6.61	78.28	47.18	45.80	80.95	79.49	78.41	71.59	47.03	66.09
C	SVD-LLM	80%	5.73	26.30	28.07	26.00	37.83	49.73	26.07	51.14	34.08	34.90
	KFAC-SVD*	80%	5.73	27.10	27.65	25.80	37.83	50.22	26.99	50.43	34.90	35.12
Q	SpinQuant	W3A8	5.61	77.74	48.81	43.00	81.04	79.11	76.01	70.09	44.88	65.09
	SERQ	W3A8	5.71	76.05	46.08	42.60	77.80	78.67	75.42	71.27	44.47	64.05
	FlatQuant	W3A8	5.61	76.47	46.08	44.00	78.72	79.16	75.73	69.93	46.52	64.58
C+Q	**CoCaQ (Eq.(3))**	5%+W3A8	5.41	75.76	47.01	43.40	79.08	77.86	76.36	70.64	44.88	64.37
	CoCaQ (Eq.(3))	10%+W3A8	5.17	76.60	45.99	43.80	79.94	77.80	75.42	70.48	44.83	64.36
	CoCaQ (Eq.(8))	5%+W3A8	5.41	77.48	45.05	45.00	78.23	79.27	75.04	70.88	45.50	64.56
	CoCaQ (Eq.(8))	10%+W3A8	5.17	77.23	47.01	45.20	77.98	78.73	76.41	73.16	44.01	64.97
C	SVD-LLM	85%	4.47	26.22	27.22	24.60	37.83	49.40	25.89	49.41	34.39	34.37
	KFAC-SVD*	85%	4.47	26.94	28.07	25.20	37.83	49.51	26.27	49.96	34.80	34.82
Q	SpinQuant	W2A8	4.02	49.49	25.34	28.80	62.78	63.33	47.20	57.85	37.56	46.54
	SERQ	W2A8	4.09	24.92	28.33	25.20	48.59	48.20	25.68	49.17	34.49	35.57
	FlatQuant	W2A8	4.02	26.09	28.07	24.00	38.04	49.89	25.65	50.43	33.73	34.49
C+Q	**CoCaQ (Eq.(3))**	5%+W2A8	3.89	45.16	25.85	32.80	61.25	62.68	41.70	55.80	36.69	45.24
	CoCaQ (Eq.(3))	10%+W2A8	3.73	46.84	26.37	30.20	61.83	61.81	40.53	52.49	36.69	44.60
	CoCaQ (Eq.(8))	20%+W3A8	4.62	71.51	39.59	40.40	73.00	74.37	65.47	64.25	40.43	58.63
	CoCaQ (Eq.(8))	5%+W2A8	3.89	52.95	27.22	33.40	64.71	65.23	46.21	57.46	37.62	48.10
	CoCaQ (Eq.(8))	10%+W2A8	3.73	57.95	28.75	36.00	64.59	65.51	51.14	57.14	38.69	49.97

Table 11: Evaluation results for LLaMA-3-8B across quantization methods and settings.

Type	Method	Setting	Memory	Arc_e	Arc_c	Openb.	BoolQ	PIQA	HellaS.	WinoG.	SIQA	CS Avg
-	Baseline	FP	16.10	80.05	53.50	45.00	80.92	80.74	79.15	74.11	46.88	67.54
C	SVD-LLM	75%	5.59	25.51	26.11	30.80	62.17	50.82	26.37	49.33	33.88	38.12
	KFAC-SVD*	75%	5.59	28.07	26.62	27.20	37.83	51.36	25.99	49.72	35.67	35.31
Q	SpinQuant	W4A8	5.69	80.35	53.75	44.60	79.17	79.98	78.17	73.40	46.93	67.04
	SERQ	W4A8	5.79	78.07	50.85	43.80	80.64	79.76	77.75	72.45	46.72	66.26
	FlatQuant	W4A8	5.69	79.25	53.07	44.00	79.48	79.71	77.94	72.53	45.65	66.45
C+Q	**CoCaQ (Eq.(3))**	5%+W4A8	5.53	79.76	53.24	45.20	80.92	80.25	78.84	73.32	45.45	67.12
	CoCaQ (Eq.(3))	10%+W4A8	5.32	68.14	41.47	41.00	70.43	75.73	70.15	68.03	42.84	59.72
	CoCaQ (Eq.(8))	5%+W4A8	5.53	78.75	52.65	45.40	79.97	80.36	77.57	74.74	46.21	66.96
	CoCaQ (Eq.(8))	10%+W4A8	5.32	79.17	50.51	45.00	72.45	79.05	73.73	71.59	44.42	64.49
C	SVD-LLM	80%	4.82	25.88	26.96	30.00	37.86	51.52	25.94	51.70	33.32	35.40
	KFAC-SVD*	80%	4.82	27.86	27.13	27.60	37.86	51.03	26.54	51.38	35.52	35.62
Q	SpinQuant	W3A8	4.81	75.63	46.67	41.80	79.72	78.29	73.89	71.43	44.88	64.04
	SERQ	W3A8	4.87	62.54	39.51	38.80	63.46	74.65	70.55	66.77	43.09	57.42
	FlatQuant	W3A8	4.81	73.40	44.62	41.00	74.46	78.18	73.63	71.11	44.27	62.59
C+Q	**CoCaQ (Eq.(3))**	5%+W3A8	4.67	73.82	45.56	43.60	79.54	79.00	74.76	70.09	43.96	63.79
	CoCaQ (Eq.(3))	10%+W3A8	4.54	62.42	35.67	39.20	60.64	74.16	64.29	64.09	40.07	55.07
	CoCaQ (Eq.(8))	5%+W3A8	4.67	74.03	47.53	44.60	76.82	78.24	73.83	71.90	45.24	64.02
	CoCaQ (Eq.(8))	10%+W3A8	4.54	73.65	43.52	42.40	73.06	76.28	67.72	68.11	42.89	60.95
C	SVD-LLM	85%	4.19	25.93	26.62	29.80	44.59	51.36	26.36	50.43	33.21	36.04
	KFAC-SVD*	85%	4.19	27.65	26.02	27.80	37.92	50.87	26.28	49.49	34.80	35.10
Q	SpinQuant	W2A8	3.95	42.63	22.95	26.40	60.46	58.38	36.15	53.99	34.80	41.97
	SERQ	W2A8	3.94	23.74	26.28	28.00	45.60	52.23	26.27	49.72	33.73	35.70
	FlatQuant	W2A8	3.95	25.34	28.33	27.20	43.03	51.20	25.67	49.49	33.11	35.42
C+Q	**CoCaQ (Eq.(3))**	5%+W2A8	3.82	38.80	21.84	28.80	58.13	56.47	37.36	52.25	35.52	41.15
	CoCaQ (Eq.(3))	10%+W2A8	3.76	33.46	21.16	25.80	48.59	53.75	30.05	49.72	34.39	37.12
	CoCaQ (Eq.(8))	20%+W3A8	4.28	52.06	25.77	32.60	42.97	61.26	36.29	52.72	35.16	42.35
	CoCaQ (Eq.(8))	5%+W2A8	3.82	24.66	26.11	29.20	53.70	52.01	35.81	52.72	34.19	38.55
	CoCaQ (Eq.(8))	10%+W2A8	3.76	32.32	20.90	27.40	46.21	53.59	29.43	50.28	34.75	36.86

Table 12: Evaluation results for Qwen-3-8B across quantization methods and settings.

Type	Method	Setting	Memory	Arc_e	Arc_c	Openb.	BoolQ	PIQA	HellaS.	WinoG.	SIQA	CS Avg
-	Baseline	FP	16.40	81.57	57.08	42.80	83.03	79.11	78.60	71.90	54.86	68.62
C	SVD-LLM	75%	5.96	27.27	25.34	24.80	37.83	52.18	27.14	50.51	33.88	34.87
	KFAC-SVD*	75%	5.96	29.67	24.23	29.40	37.83	53.05	27.85	51.78	34.24	36.01
Q	SpinQuant	W4A8	6.06	82.07	56.91	41.80	84.25	79.65	77.24	72.38	54.15	68.56
	SERQ	W4A8	6.17	80.39	54.27	43.60	80.67	79.22	77.80	71.74	51.84	67.44
	FlatQuant	W4A8	6.06	81.40	56.23	41.40	81.77	80.09	77.60	72.77	52.66	67.99
C+Q	**CoCaQ (Eq.(3))**	5%+W4A8	5.92	82.11	57.59	44.00	83.49	79.60	77.30	72.22	53.74	68.75
	CoCaQ (Eq.(3))	10%+W4A8	5.71	81.52	58.53	43.00	81.38	78.45	75.38	66.93	50.82	67.00
	CoCaQ (Eq.(8))	5%+W4A8	5.92	83.33	58.96	42.40	83.67	79.43	77.33	71.90	53.12	68.77
	CoCaQ (Eq.(8))	10%+W4A8	5.71	80.05	56.14	41.80	83.58	79.76	76.75	70.88	51.64	67.57
C	SVD-LLM	80%	5.27	26.01	24.83	24.40	37.83	50.44	26.31	48.54	33.42	33.97
	KFAC-SVD*	80%	5.27	27.69	24.83	26.40	37.83	51.36	27.59	51.46	35.01	35.27
Q	SpinQuant	W3A8	5.20	80.35	55.55	40.60	74.19	78.56	74.34	71.19	50.15	65.62
	SERQ	W3A8	5.27	79.46	50.51	38.40	79.20	75.57	72.33	67.48	47.85	63.85
	FlatQuant	W3A8	5.20	78.70	52.90	41.60	79.97	78.40	73.50	71.67	52.20	66.12
C+Q	**CoCaQ (Eq.(3))**	5%+W3A8	5.06	80.01	54.86	40.60	80.61	78.67	73.89	71.11	48.06	65.98
	CoCaQ (Eq.(3))	10%+W3A8	4.93	76.94	49.15	40.20	79.69	76.22	72.11	67.32	47.49	63.64
	CoCaQ (Eq.(8))	5%+W3A8	5.06	78.75	52.90	43.00	78.62	77.31	73.48	71.74	50.26	65.76
	CoCaQ (Eq.(8))	10%+W3A8	4.93	70.62	47.44	41.40	77.89	76.82	72.09	70.09	45.96	62.79
C	SVD-LLM	85%	4.57	26.22	25.00	24.40	37.83	50.49	25.80	47.83	33.98	33.94
	KFAC-SVD*	85%	4.57	26.01	26.11	26.40	37.83	51.52	26.97	50.51	33.83	34.90
Q	SpinQuant	W2A8	4.33	64.10	35.49	35.40	62.48	69.10	54.02	60.54	41.10	52.78
	SERQ	W2A8	4.37	25.13	25.26	26.40	43.91	51.41	26.08	51.62	33.98	35.47
	FlatQuant	W2A8	4.33	24.24	25.85	27.60	41.01	47.71	25.81	49.49	33.16	34.36
C+Q	**CoCaQ (Eq.(3))**	5%+W2A8	4.21	64.56	33.02	31.20	67.22	67.25	50.81	60.14	33.88	51.01
	CoCaQ (Eq.(3))	10%+W2A8	4.15	25.80	26.79	30.20	62.29	51.63	44.95	54.85	33.78	41.29
	CoCaQ (Eq.(8))	20%+W3A8	4.66	62.29	39.76	36.60	62.91	71.65	60.28	62.83	38.13	54.30
	CoCaQ (Eq.(8))	5%+W2A8	4.21	45.83	26.79	30.20	61.16	60.72	39.54	53.67	37.26	44.40
	CoCaQ (Eq.(8))	10%+W2A8	4.15	36.74	23.29	30.00	62.23	56.91	33.26	50.99	35.77	41.15

Table 13: Evaluation results for Qwen-3-14B across quantization methods and settings.

Type	Method	Setting	Memory	Arc_e	Arc_c	Openb.	BoolQ	PIQA	HellaS.	WinoG.	SIQA	CS Avg
-	Baseline	FP	29.50	83.42	59.13	48.60	86.91	82.05	81.35	73.88	55.73	71.38
C	SVD-LLM	75%	9.72	26.43	24.74	26.80	37.83	51.36	27.67	49.25	33.52	34.70
	KFAC-SVD*	75%	9.72	31.06	23.81	27.60	37.83	51.09	27.91	50.43	32.65	35.30
Q	SpinQuant	W4A8	9.92	83.92	59.73	48.00	87.06	81.34	80.56	72.69	55.58	71.11
	SERQ	W4A8	10.05	83.12	59.04	45.80	89.05	80.09	78.16	71.51	51.74	69.81
	FlatQuant	W4A8	9.92	83.21	58.28	47.20	87.19	81.88	80.65	73.40	55.94	70.97
C+Q	**CoCaQ (Eq.(3))**	5%+W4A8	9.58	83.50	59.04	47.40	86.54	81.23	80.86	72.06	55.37	70.75
	CoCaQ (Eq.(3))	10%+W4A8	9.25	81.94	55.89	45.80	84.86	81.28	79.45	70.48	51.84	68.94
	CoCaQ (Eq.(8))	5%+W4A8	9.58	82.03	57.59	47.60	83.91	80.69	81.11	72.93	54.35	70.03
	CoCaQ (Eq.(8))	10%+W4A8	9.25	83.50	59.90	47.60	82.54	80.90	81.03	73.64	54.50	70.45
C	SVD-LLM	80%	8.39	25.34	24.57	26.80	37.83	51.25	26.33	49.49	33.32	34.37
	KFAC-SVD*	80%	8.39	25.88	26.19	28.40	37.86	50.38	26.79	49.25	31.83	34.57
Q	SpinQuant	W3A8	8.27	83.21	59.13	44.80	86.06	80.25	78.02	72.93	48.67	69.13
	SERQ	W3A8	8.37	78.07	51.28	40.80	86.18	76.88	73.05	68.35	47.03	65.21
	FlatQuant	W3A8	8.27	82.79	58.02	44.40	86.18	80.03	78.38	71.67	49.54	68.88
C+Q	**CoCaQ (Eq.(3))**	5%+W3A8	8.06	80.43	53.07	44.60	86.18	79.38	77.96	69.69	51.59	67.86
	CoCaQ (Eq.(3))	10%+W3A8	7.76	79.59	52.30	44.80	81.65	79.49	76.55	70.56	50.36	66.91
	CoCaQ (Eq.(8))	5%+W3A8	8.06	81.90	57.34	47.80	85.23	80.30	79.19	73.32	52.76	69.73
	CoCaQ (Eq.(8))	10%+W3A8	7.76	80.22	58.28	46.80	80.73	79.22	78.34	72.61	49.90	68.26
C	SVD-LLM	85%	7.07	26.18	25.77	27.00	37.83	51.74	26.21	49.41	32.29	34.55
	KFAC-SVD*	85%	7.07	24.79	26.45	25.00	37.83	50.16	26.15	50.43	32.45	34.16
Q	SpinQuant	W2A8	6.61	71.84	42.66	37.40	68.90	72.25	61.39	65.51	41.40	57.67
	SERQ	W2A8	6.68	25.38	24.49	26.40	45.32	52.07	25.74	49.57	32.70	35.21
	FlatQuant	W2A8	6.61	25.97	26.54	29.00	41.41	50.82	26.74	50.20	33.32	35.50
C+Q	**CoCaQ (Eq.(3))**	5%+W2A8	6.45	72.56	42.75	36.40	62.45	73.07	60.81	65.27	42.43	56.97
	CoCaQ (Eq.(3))	10%+W2A8	6.27	59.05	33.19	31.20	64.01	67.08	48.65	58.48	35.88	49.69
	CoCaQ (Eq.(8))	20%+W3A8	7.24	61.03	40.96	39.80	79.14	73.45	64.76	69.53	43.24	58.99
	CoCaQ (Eq.(8))	5%+W2A8	6.45	67.85	39.25	35.00	69.36	68.61	59.88	64.01	40.07	55.50
	CoCaQ (Eq.(8))	10%+W2A8	6.27	54.50	30.29	30.40	62.23	61.81	49.70	58.88	34.34	47.77

A.11 COMPRESSION-QUANTIZATION COUPLING

The following tables compare joint compression–quantization methods on LLaMA-2-7B and Qwen3-8B.

Table 14: Compression-Quantization coupling results for LLaMA-2-7B.

Setting		Method	LLaMA-2-7B								
Quant.	**Comp.**		**Arc$_e$**	**Arc$_c$**	**Openb.**	**BoolQ**	**PIQA**	**HellaS.**	**WinoG.**	**SIQA**	**CS Avg**
FP	–	Baseline	76.26	46.08	44.00	77.92	78.94	75.93	69.22	45.96	64.29
W4A8	5%	KFAC-SVD*	75.13	41.55	42.00	75.14	78.51	73.24	68.98	44.17	62.34
		SVD-LLM	57.87	32.17	34.60	47.98	68.50	54.41	59.35	37.87	49.09
		UniQL	73.95	44.80	42.60	74.07	77.75	74.48	68.67	44.73	62.63
		CoCaQ (Eq.(8))	76.01	43.69	44.00	76.12	78.35	74.21	68.51	45.55	**63.30**
	10%	KFAC-SVD*	75.59	41.13	43.40	76.51	79.00	71.96	67.88	44.17	62.46
		SVD-LLM	54.80	28.58	34.00	42.84	66.00	52.14	60.22	37.51	47.01
		UniQL	71.68	43.60	43.80	64.65	76.93	72.98	69.22	44.98	60.98
		CoCaQ (Eq.(8))	74.87	43.34	44.20	77.19	79.27	74.45	69.77	44.83	**63.49**
W3A8	5%	KFAC-SVD*	69.11	38.31	38.40	67.13	75.79	66.58	64.96	41.10	57.67
		SVD-LLM	39.23	25.00	28.00	45.84	59.47	38.90	52.96	34.90	40.54
		UniQL	67.09	38.65	39.40	63.79	75.03	67.77	67.01	41.81	57.57
		CoCaQ (Eq.(8))	72.31	40.02	40.80	70.98	77.20	71.13	66.54	42.84	**60.23**
	10%	KFAC-SVD*	67.09	33.70	35.20	68.65	73.94	63.13	60.69	40.38	55.35
		SVD-LLM	36.79	24.49	26.20	40.00	55.11	36.86	51.93	35.31	38.34
		UniQL	65.19	37.37	40.20	58.69	74.05	66.43	65.75	42.58	56.28
		CoCaQ (Eq.(8))	71.46	39.42	40.80	71.56	76.88	69.96	65.82	43.04	**59.87**
	15%	KFAC-SVD*	62.29	32.94	31.20	65.57	72.91	59.40	56.59	39.36	52.53
		SVD-LLM	36.15	23.04	27.00	41.13	56.09	35.94	52.09	35.57	38.38
		UniQL	62.08	37.20	37.40	58.81	69.97	60.31	63.69	39.76	53.65
		CoCaQ (Eq.(8))	69.02	37.63	39.40	66.21	73.88	66.85	63.06	40.33	**57.05**
	20%	KFAC-SVD*	60.10	32.76	30.40	53.73	70.08	57.03	55.17	39.41	49.84
		SVD-LLM	34.55	22.10	28.20	38.69	54.03	34.74	50.28	35.82	37.30
		UniQL	58.71	31.57	37.60	48.20	69.26	57.79	62.98	40.74	50.86
		CoCaQ (Eq.(8))	63.51	32.00	37.00	63.94	71.33	57.56	60.93	37.92	**53.02**

Table 15: Compression-Quantization coupling results for Qwen3-8B.

Setting		Method	Qwen3-8B								
Quant.	**Comp.**		**Arc$_e$**	**Arc$_c$**	**Openb.**	**BoolQ**	**PIQA**	**HellaS.**	**WinoG.**	**SIQA**	**CS Avg**
FP	–	Baseline	81.57	57.08	42.80	83.03	79.11	78.60	71.90	54.86	68.62
W4A8	5%	KFAC-SVD*	79.17	53.41	37.60	79.51	79.49	72.73	65.27	43.30	63.81
		SVD-LLM	67.21	44.28	39.60	69.48	73.72	65.98	66.38	42.99	58.71
		UniQL	82.66	60.15	42.60	82.91	77.91	77.81	71.74	51.02	68.35
		CoCaQ (Eq.(8))	83.33	58.96	42.40	83.67	79.43	77.33	71.90	53.12	**68.77**
	10%	KFAC-SVD*	70.16	41.47	38.60	75.93	72.52	70.31	62.67	42.43	59.26
		SVD-LLM	63.17	42.32	38.60	63.46	71.60	64.56	64.72	40.58	56.13
		UniQL	62.04	39.16	36.20	59.14	72.31	67.88	61.40	41.91	55.01
		CoCaQ (Eq.(8))	80.05	56.14	41.80	83.58	79.76	76.75	70.88	51.64	**67.57**
W3A8	5%	KFAC-SVD*	53.45	30.03	32.60	71.56	60.61	61.42	58.41	42.02	51.26
		SVD-LLM	52.99	33.45	33.00	60.34	64.31	48.33	57.14	38.13	48.46
		UniQL	74.54	47.44	41.20	67.19	74.97	71.23	65.04	43.35	60.62
		CoCaQ (Eq.(8))	78.75	52.90	43.00	78.62	77.31	73.48	71.74	50.26	**65.76**
	10%	KFAC-SVD*	50.97	30.97	29.80	66.57	62.51	58.46	53.28	41.50	49.26
		SVD-LLM	52.36	29.78	35.40	61.50	64.85	47.56	56.75	37.72	48.24
		UniQL	46.25	29.78	34.60	50.76	62.84	54.15	52.09	36.69	45.90
		CoCaQ (Eq.(8))	70.62	47.44	41.40	77.89	76.82	72.09	70.09	45.96	**62.79**
	15%	KFAC-SVD*	49.54	29.52	27.40	64.95	62.30	52.21	55.88	41.40	47.90
		SVD-LLM	45.46	28.41	28.00	56.02	59.20	38.91	54.22	36.03	43.28
		UniQL	46.00	31.06	32.40	53.00	62.13	53.87	51.38	36.08	45.74
		CoCaQ (Eq.(8))	63.05	41.98	40.00	77.13	72.03	69.22	65.51	35.62	**58.07**
	20%	KFAC-SVD*	43.81	26.79	30.60	62.57	60.72	46.80	53.99	40.17	45.68
		SVD-LLM	44.36	26.54	27.80	52.48	59.52	36.56	50.04	35.42	41.59
		UniQL	41.29	28.41	28.40	51.35	59.25	45.46	52.72	35.82	42.84
		CoCaQ (Eq.(8))	62.29	39.76	36.60	62.91	71.65	60.28	62.83	38.13	**54.30**

A.12 COMPARISON BETWEEN TWO TYPES OF QF-SVD

The following table compares the two QF-SVD methods on LLaMA-2-7B and Qwen3-8B.

Table 16: Commonsense results of weight-based and activation-based CoCaQ under W3A8.

Model	Quant.	Comp.	Method	Commonsense Accuracy (%)								
				Arc_e	Arc_c	Openb.	BoolQ	PIQA	HellaS.	WinoG.	SIQA	CS Avg
LLaMA-2-7B	W3A8	5%	**CoCaQ (Eq. (3))**	73.02	39.59	39.20	74.01	76.61	71.12	67.56	43.30	60.55
			CoCaQ (Eq. (8))	72.31	40.02	40.80	70.98	77.20	71.13	66.54	42.84	60.23
		10%	**CoCaQ (Eq. (3))**	72.43	40.02	43.20	72.84	76.44	69.63	67.48	42.94	60.62
			CoCaQ (Eq. (8))	71.46	39.42	40.80	71.56	76.88	69.96	65.82	43.04	59.87
		15%	**CoCaQ (Eq. (3))**	66.41	37.88	40.00	68.35	74.21	66.73	62.83	41.56	57.25
			CoCaQ (Eq. (8))	69.02	37.63	39.40	66.21	73.88	66.85	63.06	40.33	57.05
		20%	**CoCaQ (Eq. (3))**	58.54	33.36	35.80	65.57	69.86	56.95	60.77	38.79	52.46
			CoCaQ (Eq. (8))	63.51	32.00	37.00	63.94	71.33	57.56	60.93	37.92	53.02
Qwen3-8B	W3A8	5%	**CoCaQ (Eq. (3))**	80.01	54.86	40.60	80.61	78.67	73.89	71.11	48.06	65.98
			CoCaQ (Eq. (8))	78.75	52.90	43.00	78.62	77.31	73.48	71.74	50.26	65.76
		10%	**CoCaQ (Eq. (3))**	76.94	49.15	40.20	79.69	76.22	72.11	67.32	47.49	63.64
			CoCaQ (Eq. (8))	70.62	47.44	41.40	77.89	76.82	72.09	70.09	45.96	62.79
		15%	**CoCaQ (Eq. (3))**	55.35	36.26	35.00	55.50	68.66	59.65	53.91	37.87	50.28
			CoCaQ (Eq. (8))	63.05	41.98	40.00	77.13	72.03	69.22	65.51	35.62	58.07
		20%	**CoCaQ (Eq. (3))**	27.65	24.32	25.80	47.58	50.76	28.16	49.57	33.16	35.88
			CoCaQ (Eq. (8))	62.29	39.76	36.60	62.91	71.65	60.28	62.83	38.13	54.30

A.13 STABILITY RESULTS FOR R_* INITIALIZATION

The following tables report variation across random seeds for QR and Hadamard rotations under the same compression and quantization setting.

Table 17: Seed sensitivity of QR and Hadamard rotations for LLaMA-2-7B under 5% compression and W3A8 quantization.

Seed	QR			Hadamard		
	Wiki2↓	MMLU↑	CS↑	Wiki2↓	MMLU↑	CS↑
42	6.03	39.84	60.23	6.02	38.06	60.53
233	6.02	38.31	60.19	6.03	41.01	60.18
456	6.04	39.33	60.04	6.04	37.69	60.11
666	6.05	38.79	60.27	6.03	40.59	60.35
789	6.04	40.57	60.36	6.04	36.04	60.29
Average	6.04	39.37	60.22	6.04	38.68	60.29
Std.	0.01	0.88	0.12	0.01	2.09	0.16

Table 18: Per-seed commonsense results with QR under the same setting.

Seed	QR								
	Arc_e	Arc_c	Openb.	BoolQ	PIQA	HellaS.	WinoG.	SIQA	CS Avg↑
42	72.31	40.02	40.80	70.98	77.20	71.13	66.54	42.84	60.23
233	72.26	39.85	40.40	70.31	76.55	70.99	67.48	43.71	60.19
456	71.59	40.36	40.40	71.50	76.88	70.94	65.82	42.84	60.04
666	72.22	40.44	40.60	71.16	76.88	70.22	66.93	43.71	60.27
789	72.60	39.93	40.00	72.14	76.66	70.89	66.77	43.91	60.36

Table 19: Per-seed commonsense results with Hadamard rotation under the same setting.

Seed	Hadamard								
	Arc_e	Arc_c	Openb.	BoolQ	PIQA	HellaS.	WinoG.	SIQA	CS Avg↑
42	72.14	40.53	41.60	71.53	76.17	70.86	67.64	43.76	60.53
233	71.17	40.19	40.00	71.47	76.01	71.48	67.48	43.65	60.18
456	72.64	39.76	40.20	72.69	75.79	71.29	65.59	42.89	60.11
666	72.90	40.87	40.60	71.35	76.39	71.11	66.46	43.14	60.35
789	71.97	40.96	40.40	71.59	76.17	71.24	66.38	43.60	60.29