Assigning Distinct Roles to Quantized and Low-Rank Matrices Toward Optimal Weight Decomposition

Yoonjun Cho[1] **Soeun Kim**[2] **Dongjae Jeon**[1] **Kyelim Lee**[2] **Beomsoo Lee**[3] **Albert No**[2†]

[1]Department of Computer Science, Yonsei University
[2]Department of Artificial Intelligence, Yonsei University
[3]Department of Electronic & Electrical Convergence Engineering, Hongik University

Abstract

Decomposing weight matrices into quantization and low-rank components ($\mathbf{W} \approx \mathbf{Q}+\mathbf{LR}$) is a widely used technique for compressing large language models (LLMs). Existing joint optimization methods iteratively alternate between quantization and low-rank approximation. However, these methods tend to prioritize one component at the expense of the other, resulting in suboptimal decompositions that fail to leverage each component's unique strengths. In this work, we introduce Outlier-Driven Low-Rank Initialization (ODLRI), which assigns low-rank components the specific role of capturing activation-sensitive weights. This structured decomposition mitigates outliers' negative impact on quantization, enabling more effective balance between quantization and low-rank approximation. Experiments on Llama2 (7B, 13B, 70B), Llama3-8B, and Mistral-7B demonstrate that incorporating ODLRI into the joint optimization framework consistently reduces activation-aware error, minimizes quantization scale, and improves perplexity and zero-shot accuracy in low-bit settings.

1 Introduction

Quantization (Polino et al., 2018; Jacob et al., 2018; Nagel et al., 2021) and weight matrix factorization (Golub et al., 1987; Saha et al., 2023) are two widely used techniques for compressing large language models (LLMs) to enable efficient inference on resource-constrained hardware. Post-training quantization (PTQ) reduces model size and computational cost by mapping high-precision weights to lower-bit representations (Tao et al., 2022; Bai et al., 2022; Dettmers et al., 2022; Shao et al., 2024), while matrix factorization approximates weight matrices with compact factored representations (Li et al., 2023; Gao et al., 2024b). Recent joint optimization approaches combine these methods to achieve extreme compression, representing weights as the sum of a quantization matrix and a low-rank component: $\mathbf{W} \approx \mathbf{Q} + \mathbf{LR}$ (Saha et al., 2024; Li et al., 2024b; Guo et al., 2024).

Joint optimization approaches minimize representation error by iteratively alternating between quantization and low-rank approximation. These methods typically adopt either a quantize-first strategy (Saha et al., 2024; Li et al., 2024b) or a low-rank-first strategy (Guo et al., 2024). While these differ in iteration ordering, they can be equivalently understood as distinct initialization choices for the low-rank components: the quantize-first approach initializes $\mathbf{LR}$ to zero, while the low-rank-first approach initializes them using factorized weights.

Critically, existing methods treat these different orderings merely as preparatory steps, assuming iterative updates will naturally converge to optimal solutions. However, viewing this process through the lens of initialization reveals an underexplored aspect of how initialization strategies might fundamentally affect weight decomposition quality.

In this work, we investigate the role of initialization in joint optimization. Our analysis shows that different initializations lead to distinct solution spaces, with components maintaining persistent roles throughout optimization. Quantize-first methods treat low-rank components as error correction terms, while low-rank-first methods preserve them as the primary weight representation. This finding highlights that initialization fundamentally determines the role assignment between quantization and low-rank components, raising a key question:

> *What is the optimal initialization strategy for decomposing weights into quantized and low-rank matrices?*

Recent works have shown that quantization errors are pronounced for weights associated with activation outliers, as extreme activations amplify weight sensitivity (Dettmers et al., 2023b; Lin et al.,

†Correspondence to: Albert No <albertno@yonsei.ac.kr>

Findings of the Association for Computational Linguistics: ACL 2025, pages 14453–14470

2024; Lee et al., 2024; Huang et al., 2024; Kim et al., 2024). Building on this insight, we introduce Outlier-Driven Low-Rank Initialization (ODLRI), which assigns a specific role to the low-rank component to capture these salient weights while using the quantized matrix to express the residuals. By handling outlier-sensitive weights through the low-rank component, our approach stabilizes quantization and enables more precise weight decomposition.

Through extensive experiments on Llama2 (7B, 13B, 70B) (Touvron et al., 2023), Llama3-8B (Dubey et al., 2024), and Mistral-7B (Jiang et al., 2023), we demonstrate that incorporating ODLRI into the joint optimization framework consistently improves perplexity and zero-shot accuracy across extreme low-bit settings. Our analysis shows that ODLRI reduces activation-aware error, minimizes quantization scale, and enhances model performance, highlighting the importance of structured initialization in low-bit quantization. These results present a principled approach for stable and efficient LLM compression, and provide new insights into the effective decomposition of quantization and low-rank matrices.

We summarize our contributions as follows:

- We propose a unified framework for expressing iterative joint optimization algorithms by introducing the concept of initialization of low-rank components.
- We analyze the impact of initialization on iterative joint optimization algorithms, revealing the suboptimality of conventional approaches.
- We propose Outlier-Driven Low-Rank Initialization (ODLRI), which assigns a specific role to the low-rank component $\mathbf{LR}$ to capture salient weights while using the quantization matrix $\mathbf{Q}$ for the remaining weights.
- Comprehensive experiments demonstrate that incorporating ODLRI reduces activation-aware error, minimizes quantization scale, and improves perplexity and zero-shot accuracy.

2 Preliminaries

2.1 Post-Training Quantization

Early post training quantization (PTQ) methods (Jacob et al., 2018; Nagel et al., 2021; Dettmers et al., 2022; Shao et al., 2024) mainly focus on minimizing the direct quantization error for a given weight matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$, optimizing:

$$\arg\min_{\mathbf{Q}} \|\mathbf{W} - \mathbf{Q}\|_{\mathrm{F}}^2,$$

where $\mathbf{Q}$ is obtained by rounding weights to the nearest discrete values (Banner et al., 2019; Stock et al., 2020; Wu et al., 2020). However, this naive rounding approach often results in significant accuracy degradation. This problem is particularly pronounced in large-scale LLMs, where even small perturbations in weights can propagate across layers, gradually accumulating and worsening errors.

To mitigate this issue, activation-aware PTQ methods (Dong et al., 2019; Nagel et al., 2020; Li et al., 2021, 2024a) incorporate a calibration dataset $\mathbf{X} \in \mathbb{R}^{n\times d}$ to account for the interaction between weight quantization and activations:

$$\arg\min_{\mathbf{Q}} \|(\mathbf{W} - \mathbf{Q})\mathbf{X}\|_{\mathrm{F}}^2.$$

This formulation ensures that weight quantization preserves the statistical behavior of activations, and leads to improved model performance. OPTQ (Frantar et al., 2023) refines activation-aware PTQ by introducing error feedback, which reduces cumulative quantization errors for effective low-bit quantization. QuIP (Chee et al., 2023) and QuIP# (Tseng et al., 2024) further improve robustness through incoherence processing, applying orthogonal transformations to $\mathbf{W}$ and its local Hessian to reduce correlations, making 2-bit quantization more effective.

In addition, recent studies show that effectively controlling a small portion of activation outlier-sensitive weights can significantly enhance quantization performance. SpQR (Dettmers et al., 2023b) retains these critical salient weights in higher precision, while AWQ employs per-channel scaling to protect them, effectively mitigating the adverse effects of activation outliers during quantization.

2.2 Weight Matrix Factorization

Matrix factorization decomposes a matrix into low-rank components for efficient representation and computation (Golub et al., 1987; Saha et al., 2023). Recently, it has been applied to LLM compression by approximating weight matrices with low-rank representations (Li et al., 2023; Gao et al., 2024b). Given a weight matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$, these methods decompose it into lower-dimensional matrices, $\mathbf{L} \in \mathbb{R}^{m\times r}$ and $\mathbf{R} \in \mathbb{R}^{r\times n}$, by minimizing:

$$\arg\min_{\mathbf{L},\mathbf{R}} \|\mathbf{W} - \mathbf{LR}\|_{\mathrm{F}}^2.$$

By reducing the rank r, low-rank decomposition, such as Singular Value Decomposition (SVD) effectively reduces storage and computational costs while preserving key weight structures.

Activation-aware factorization further incorporates activation statistics to refine weight representation, keeping the decomposition aligned with the model's actual computational behavior. ASVD (Yuan et al., 2023) introduces a diagonal scaling matrix to normalize activations, improving numerical stability, while SVD-LLM (Wang et al., 2025) applies truncation-aware data whitening via Cholesky decomposition.

2.3 Quantization Error Reconstruction

Empirical studies suggest that quantization errors $\mathbf{W} - \mathbf{Q}$ often exhibit a low-rank structure (Yao et al., 2024). This observation has led to hybrid approaches that first quantize the weight matrix and then approximate the resulting quantization error using a low-rank term, which serves as an error compensation mechanism:

$$\mathbf{Q} = \arg\min_{\mathbf{Q}} \|\mathbf{W} - \mathbf{Q}\|_{\mathrm{F}}^2$$
$$\mathbf{L}, \mathbf{R} = \arg\min_{\mathbf{L},\mathbf{R}} \|\mathbf{W} - \mathbf{Q} - \mathbf{L}\mathbf{R}\|_{\mathrm{F}}^2.$$

Here, $\mathbf{Q}$ represents the quantized weight matrix, forming the major representation of $\mathbf{W}$, while $\mathbf{LR}$ provide a low-rank approximation of the residual quantization error (Liu et al., 2024).

ZeroQuant-V2 (Yao et al., 2024) estimates $\mathbf{L}$ and $\mathbf{R}$ by applying SVD to the residual error $\mathbf{W} - \mathbf{Q}$. Subsequent methods further refine this process by incorporating activation-aware adjustments: LQER (Zhang et al., 2024) introduces a diagonal scaling matrix derived from activations, while QERA (Zhang et al., 2025) improves error reconstruction by leveraging an input-space autocorrelation matrix, and provides theoretical guarantees for enhanced performance.

Unlike Q-LoRA (Dettmers et al., 2023a) and other quantization-based parameter-efficient fine-tuning (Q-PEFT) methods that focus on weight representations optimized for fine-tuning, quantization error reconstruction methods solely aim for efficient weight representation.

2.4 Jointly Optimized Quantization and Low Rank Approximation

A natural extension of quantization error reconstruction is to jointly optimize both the quantized component $\mathbf{Q}$ and the low-rank component $\mathbf{LR}$, leading to the following formulation:

$$\mathbf{Q}, \mathbf{L}, \mathbf{R} = \arg\min_{\mathbf{Q},\mathbf{L},\mathbf{R}} \|(\mathbf{W} - \mathbf{Q} - \mathbf{L}\mathbf{R})\|_{\mathrm{F}}^2.$$

Unlike the two-stage approach that applies quantization followed by low-rank error correction once, joint optimization methods often iteratively alternate between quantization and low-rank approximation, which generally leads to better performance. These methods typically adopt either a quantize-first strategy (Saha et al., 2024; Li et al., 2024b) or a low-rank-first strategy (Guo et al., 2024), distinguished by their iteration ordering.

Rather than distinguishing them by iteration sequences, we propose interpreting these approaches through their initialization of low-rank components. Specifically, the quantize-first approach initializes $\mathbf{LR}$ to zero, whereas the low-rank-first approach initializes them to factorized weights. Our perspective enables us to reframe joint optimization methods within a unified view in Algorithm 1.

Algorithm 1 Joint $\mathbf{Q} + \mathbf{LR}$ Optimization

Input: Pretrained weight $\mathbf{W}$, Num of iterations T
Output: $\mathbf{Q}_T, \mathbf{L}_T, \mathbf{R}_T$

$\mathbf{L}_0, \mathbf{R}_0 \leftarrow$ **Initialize**
for $t = 1$ to T **do**
  $\mathbf{Q}_t \leftarrow$ **Quantize**$(\mathbf{W} - \mathbf{L}_{t-1}\mathbf{R}_{t-1})$
  $\mathbf{L}_t, \mathbf{R}_t \leftarrow$ **LRApprox**$(\mathbf{W} - \mathbf{Q}_t)$
end for
Return: $\widehat{\mathbf{W}} = \mathbf{Q}_T + \mathbf{L}_T \cdot \mathbf{R}_T$

Through Algorithm 1, we gain deeper insights into the joint optimization procedure. This framework not only allows us to explore algorithmic variations by examining different initialization strategies but also suggests the potential role of initialization choices in weight decomposition.

CALDERA (Saha et al., 2024), a weight-only PTQ method jointly optimizes $\mathbf{Q}$, $\mathbf{L}$, and $\mathbf{R}$ in an activation-aware manner, has shown significant improvements in quantization quality, particularly in extreme low-bit settings. Despite its outstanding performance, CALDERA solely examines its approach by initializing low-rank components to zero, neglecting the potential impact of initialization on performance. Within this CALDERA framework, we further concentrate on how different initialization strategies for the low-rank component affect performance, aiming to shed light on its role in the overall optimization process.

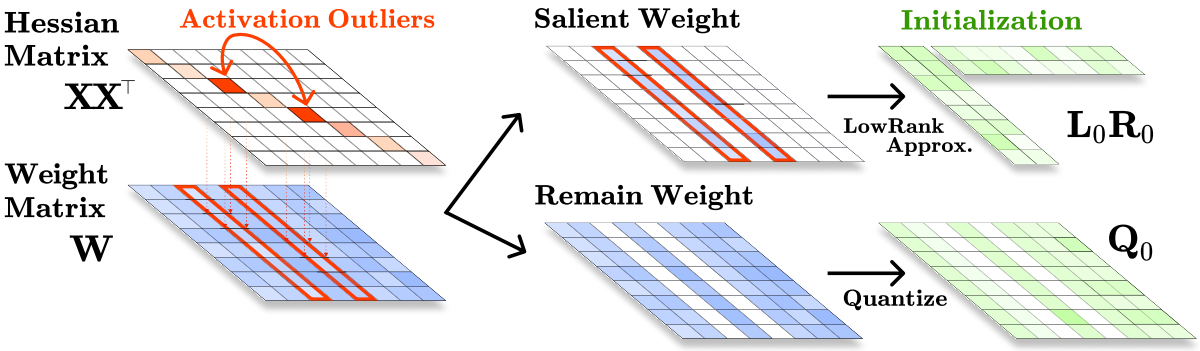


Figure 1: **Outlier-Driven Low-Rank Initialization (ODLRI) Framework.** ODLRI decomposes the weight matrix $\mathbf{W}$ by first identifying salient weights, corresponding to activation outliers, using the diagonal of the Hessian. These salient weights are then approximated via low-rank decomposition, producing $\mathbf{L}_0$ and $\mathbf{R}_0$, while the remaining weights are quantized. This decomposition serves as initialization for the iterative joint optimization of $\mathbf{Q} + \mathbf{LR}$.

3 Rethinking Joint Q+LR Optimization

3.1 Dependence on LR Initialization

To examine the role of initialization in joint optimization, we evaluate CALDERA (Saha et al., 2024) under two distinct initialization strategies for $\mathbf{LR}$: zero initialization and matrix factorization-based initialization. For analyzing the contribution of quantized representation and low-rank components to $\mathbf{WX}$, we measure $\|\mathbf{QX}\|$ and $\|\mathbf{LRX}\|$ at both the initial and final stage of iteration.

Surprisingly, as shown in Table 1, when $\mathbf{L}, \mathbf{R}$ are initialized to zero, we observe that $\mathbf{Q}$ persistently reconstructs $\mathbf{W}$, while $\mathbf{LR}$ serves as a residual correction, closely resembling quantization error reconstruction. Conversely, initializing $\mathbf{L}, \mathbf{R}$ with a matrix factorization of $\mathbf{W}$ leads to a reversed role assignment, where $\mathbf{LR}$ captures most of $\mathbf{W}$, and $\mathbf{Q}$ quantizes the residuals throughout the iteration.

LR Initialization	**0**		LRApprox($\mathbf{W}$)	
$\setminus$	$\|\mathbf{QX}\|$	$\|\mathbf{LRX}\|$	$\|\mathbf{QX}\|$	$\|\mathbf{LRX}\|$
First Iteration	0.999	0.014	0.158	0.915
Last Iteration	0.961	0.073	0.401	0.664

Table 1: Effect of $\mathbf{LR}$ Initialization in CALDERA. Activation norms $\|\mathbf{QX}\|$ and $\|\mathbf{LRX}\|$ are reported at the first and last iterations for the Layer 1 Key Projection matrix of Llama2-7B over 15 iterations. Norms are normalized by $\|\mathbf{WX}\|$ (i.e., $\|\mathbf{LRX}\|/\|\mathbf{WX}\|$). More results are provided in Appendix C.4.

This finding reveals that joint optimization outcomes are highly sensitive to initialization choices, which ultimately determine whether quantization or matrix factorization dominates the final representation. While existing methods have defaulted to either zero initialization or low-rank approximation of $\mathbf{W}$ (i.e., LRApprox($\mathbf{W}$)), the optimality of these conventional approaches remains unexplored. Rather than being restricted to these established choices, we ask:

> *How should we initialize* $\mathbf{L}, \mathbf{R}$ *to achieve optimal decomposition of* $\mathbf{W}$ *into* $\mathbf{Q} + \mathbf{LR}$*?*

3.2 Outlier-Driven Low-Rank Initialization

For optimal decomposition of $\mathbf{W} \approx \mathbf{Q} + \mathbf{LR}$, we assign distinct roles to quantization and low-rank approximation based on their properties. Quantization is highly sensitive to activation outliers, as extreme activations amplify weight sensitivity, leading to discretization errors that degrade model performance (Lin et al., 2024; Dettmers et al., 2023b). In contrast, the low-rank component utilizes a product formulation of two low-bit factors, effectively yielding a higher bit representation than quantization. To leverage the enhanced representational capacity of the low-rank component, we structure $\mathbf{LR}$ to explicitly capture salient weights before quantization, rather than treating it as a post-hoc correction term. This results in salient weights being absorbed into the low-rank component, allowing $\mathbf{Q}$ to operate on a smoother, more uniform residual, ultimately improving quantization efficiency.

To explicitly assign the low-rank component to capturing salient weights, we introduce *Outlier-Driven Low-Rank Initialization (ODLRI)*, as illustrated in Figure 1. More precisely, we first decompose the activation matrix $\mathbf{X}$ into two components:

$$\mathbf{X} = \mathbf{X}_\text{o} + \mathbf{X}_\text{r},$$

where $\mathbf{X}_{\text{o}} \in \mathbb{R}^{n \times d}$ contains only the top-k activation channels (outliers) with the highest norms, the remaining channels are set to zero. These top-k activation channels are identified by analyzing the diagonal entries of the Hessian, computed as $\mathbf{H} = \mathbf{X}\mathbf{X}^\top$. The remaining activations (non-outlier components) are then captured by $\mathbf{X}_{\text{r}}$.

Although setting $k = r$ (where $\mathbf{L} \in \mathbb{R}^{m \times r}$ and $\mathbf{R} \in \mathbb{R}^{r \times n}$) would maximize the use of low-rank approximation, we intentionally choose $k < r$ to focus aggressively on outlier-related structures rather than broadly approximating the entire weight distribution. This targeted selection ensures that the low-rank component prioritizes the most activation-sensitive elements, refining the decomposition and enhancing quantization robustness.

We then initialize $\mathbf{L}$ and $\mathbf{R}$ via outlier-aware matrix factorization, focusing on high-variance activation directions that are likely to induce quantization errors. Specifically, we solve the following truncated optimization problem:

$$\mathbf{L}_0, \mathbf{R}_0 = \arg\min_{\mathbf{L},\mathbf{R}} \|(\mathbf{W} - \mathbf{L}\mathbf{R})\mathbf{H}_{\text{o}}(\mathbf{W} - \mathbf{L}\mathbf{R})^\top\|,$$

where $\mathbf{H}_{\text{o}} = \mathbf{X}_{\text{o}}\mathbf{X}_{\text{o}}^\top$ captures the covariance of outlier-sensitive channels. This objective ensures that $\mathbf{L}_0\mathbf{R}_0$ prioritizes reconstructing weight directions that interact strongly with outlier channels in the activation distribution. A detailed algorithm for solving this optimization is provided in Appendix B.1, and the specific selection of k is described in Appendix B.2.

4 Experiment

4.1 Experimental Setup

To evaluate the effectiveness of Outlier-Driven Low-Rank Initialization (ODLRI), we integrate it into CALDERA (Saha et al., 2024), an iterative joint optimization framework for extreme low-bit quantization of $\mathbf{Q}, \mathbf{L}, \mathbf{R}$. By default, CALDERA first quantizes the weight matrix and then optimizes a low-rank component, effectively treating $\mathbf{L}, \mathbf{R}$ as zero-initialized correction factors.

In contrast, ODLRI replaces this zero-initialization with an outlier-aware initialization, ensuring that salient weights are explicitly modeled in the low-rank component before quantization. We conduct a comprehensive comparison against CALDERA's standard setup to quantify the impact of $\mathbf{LR}$ initialization on final accuracy and stability.

Quantization Setup. We integrate ODLRI into CALDERA while keeping the quantization configuration largely unchanged. For **Quantize**, we use a 2-Bit quantization scheme implemented via QuIP# (Tseng et al., 2024), which employs an E8 lattice codebook for stable 2-Bit quantization. For **LRApprox**, we experiment with both 16-Bit and 4-Bit precision, where the 16-Bit setting remains unquantized, while the 4-Bit setting undergoes quantization. For 4-Bit precision, we adjust $\mathbf{L}$ and $\mathbf{R}$ via the LPLR iterative algorithm (Saha et al., 2023).

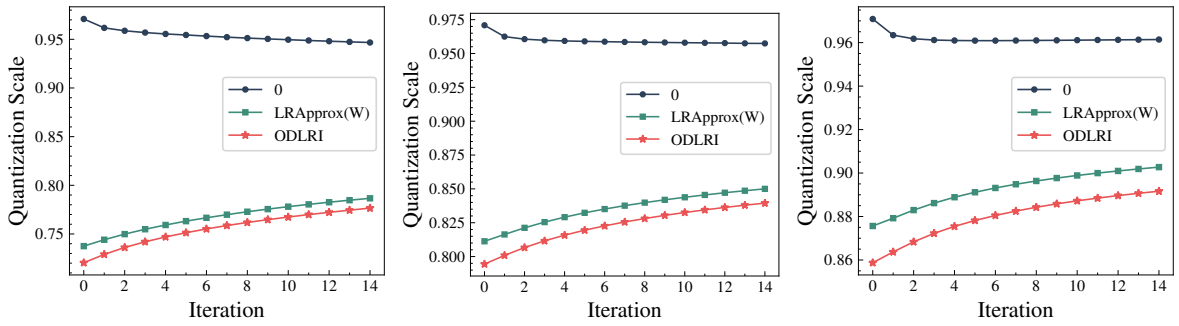
CALDERA performs outer iterations alternating between **Quantize** and **LRApprox**, along with inner iterations for the LPLR procedure. We follow CALDERA's default configuration, running 15 outer iterations, and 10 inner iterations when the $\mathbf{L}, \mathbf{R}$ components are quantized to 4-Bit.

Models. We conduct experiments on a range of large language models, including Llama2 (7B, 13B, and 70B parameters) (Touvron et al., 2023), Llama3-8B (Dubey et al., 2024), and Mistral-7B (Jiang et al., 2023).

Evaluation Metrics. We evaluate quantization performance using perplexity and zero-shot accuracy benchmarks. Perplexity is measured on the test splits of WikiText-2 (Merity et al., 2017) and C4 (Raffel et al., 2020), following each model's predefined context length (e.g., 4096 tokens for Llama2 and 8192 tokens for Llama3). Zero-shot accuracy is assessed using the EleutherAI `lm-evaluation-harness` (Gao et al., 2024a), covering a range of NLP benchmarks, including Winogrande (Sakaguchi et al., 2020), RTE (Bentivogli et al., 2009), PiQA (Bisk et al., 2020), ARC Easy (Clark et al., 2018), and ARC Challenge (Clark et al., 2018). To ensure statistical reliability, most results are averaged across two independent random seeds (Details are in Appendix A).

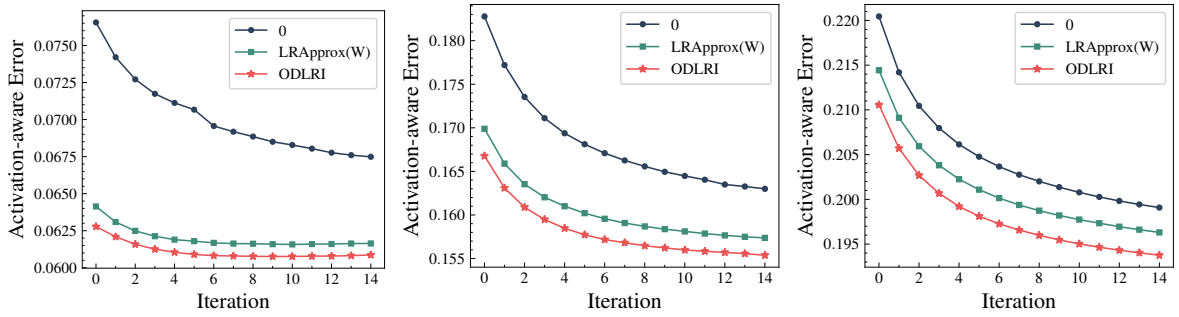
4.2 Effect of LR Initialization on Joint Q+LR Optimization

We compare three initialization methods for $\mathbf{L}, \mathbf{R}$ to evaluate their impact on performance. The first method, Zero Initialization (0), follows CALDERA's default setup, where $\mathbf{L}_0 = 0$ and $\mathbf{R}_0 = 0$. The second method, LRApprox($\mathbf{W}$), initializes $\mathbf{L}, \mathbf{R}$ using a low-rank approximation (LPLR) of $\mathbf{W}$. The third method, ODLRI, is our proposed approach, which explicitly focuses salient weights for low-rank approximation.



(a) Key projection (b) Value projection (c) Down Projection

Figure 2: **Quantization Scale across different initialization strategies**. We present the quantization scale over 15 iterations, where both $\mathbf{L}$ and $\mathbf{R}$ are quantized to 4-Bit at rank 256. The three subplots display results for the Key (left), Value (middle), and Down (right) projection layers in Layer 10 of Llama2-7B. ODLRI (red stars) consistently achieves the lowest quantization scale, highlighting its effectiveness in low-bit quantization. Additional results are provided in the Appendix C.5



(a) Key projection (b) Value projection (c) Down Projection

Figure 3: **Activation-aware Error across different initialization strategies**. We present the normalized activation-aware error $\|(\mathbf{W}-\mathbf{Q}-\mathbf{LR})\mathbf{X}\|_{\mathrm{F}}^2/\|\mathbf{WX}\|_{\mathrm{F}}^2$ over 15 iterations, where both $\mathbf{L}$ and $\mathbf{R}$ are quantized to 4-Bit at rank 256. The three subplots display results for the Key (left), Value (middle), and Down (right) projection layers in Layer 10 of Llama2-7B. ODLRI (red stars) consistently achieves the lowest error, demonstrating its effectiveness in accurately representing weights. Additional results are provided in the Appendix C.5

Quantization Scale. We first examine the quantization scale, which reflects the dynamic range of the weights and directly impacts low-bit quantization efficiency. A lower quantization scale indicates a more compact weight distribution, enabling finer representation in low-bit precision and reducing overall quantization error.

Figure 2 shows that ODLRI significantly reduces the quantization weight scale, ensuring that weights are mapped into a range more suitable for accurate quantization. In contrast, baseline methods exhibit higher scales, making quantization more challenging and increasing the risk of performance degradation. This reduction in quantization scale with ODLRI is a key factor in achieving superior performance in extreme low-bit settings.

Additional results on other layers and different projection types confirm consistent scale reduction by ODLRI (see Appendix C.5).

Activation-aware Error. To further validate that ODLRI effectively minimizes our objective, we measure the normalized activation-aware error for each layer's weight component. This metric evaluates how well the decomposition preserves activation-dependent weight structure:

$$\frac{\|(\mathbf{W}-\mathbf{Q}-\mathbf{LR})\mathbf{X}\|_{\mathrm{F}}^2}{\|\mathbf{WX}\|_{\mathrm{F}}^2}.$$

As illustrated in Figure 3, the default CALDERA configuration with zero initialization results in significantly higher activation-aware errors. Even when using a low-rank approximation of $\mathbf{W}$, the error remains consistently higher across layers compared to ODLRI. The substantial reduction in activation-aware error achieved by ODLRI confirms that it represents weight approximations more effectively. Additional results are provided in Appendix C.5.

Model	Method	Rank	Avg Bits	PPL ↓		Zero-Shot Accuracy ↑				
				Wiki2	C4	Wino	RTE	PiQA	ArcE	ArcC
7B	CALDERA	64	2.1	7.34	**9.50**	63.85	55.23	72.69	62.25	32.04
	+ODLRI	64	2.1	**7.20**	9.52	**65.04**	**62.45**	**72.91**	**65.49**	**33.70**
	CALDERA	128	2.2	6.90	9.01	**65.27**	57.58	72.80	**63.89**	**34.21**
	+ODLRI	128	2.2	**6.72**	**8.82**	64.33	**57.76**	**75.30**	62.63	33.19
	CALDERA	256	2.4	6.47	8.47	65.19	60.11	74.43	**66.35**	34.25
	+ODLRI	256	2.4	**6.33**	**8.27**	**66.42**	**61.01**	**74.67**	65.49	**36.09**
13B	CALDERA	64	2.08	5.92	**7.96**	66.37	58.84	75.27	**69.00**	**38.18**
	+ODLRI	64	2.08	**5.91**	7.97	**67.32**	**63.18**	**75.90**	65.82	37.20
	CALDERA	128	2.16	5.77	**7.71**	67.4	**62.45**	75.70	**69.65**	38.91
	+ODLRI	128	2.16	**5.73**	**7.71**	**69.69**	61.37	**76.50**	68.69	**39.32**
	CALDERA	256	2.32	5.56	7.39	**69.13**	**64.08**	75.98	70.26	39.88
	+ODLRI	256	2.32	**5.46**	**7.28**	68.19	62.63	**76.65**	**71.48**	**39.96**
70B	CALDERA	128	2.1	4.09	5.91	**75.93**	71.84	79.38	76.98	47.10
	+ODLRI	128	2.1	**4.06**	**5.89**	75.30	**72.56**	**79.98**	**78.45**	**49.06**
	CALDERA	256	2.2	3.99	5.78	75.69	**72.92**	80.41	**78.28**	49.15
	+ODLRI	256	2.2	**3.94**	**5.73**	**75.91**	70.75	**80.73**	77.96	**49.91**
Uncompressed (7B)		—	16	5.12	6.63	67.3	63.2	78.5	69.3	40.0
Uncompressed (13B)		—	16	4.57	6.05	69.5	61.7	78.8	73.2	45.6
Uncompressed (70B)		—	16	3.12	4.97	77.0	67.9	81.1	77.7	51.1

Table 2: Comparison of our method with CALDERA on zero-shot perplexities (↓) and accuracies (↑) of Llama2 models on WikiText-2 and C4. $\mathbf{Q}$ is quantized 2-Bit and $\mathbf{L}, \mathbf{R}$ are quantized 4-Bit. Lower perplexity and higher accuracy values are **bolded**.

4.3 Zero-shot Evaluation

Following the setup in the previous section, we conduct zero-shot evaluations of various models with a fixed 2-Bit $\mathbf{Q}$ component and an $\mathbf{LR}$ component set to either quantized 4-Bit or unquantized 16-Bit. The goal is to assess how applying ODLRI initialization compares against the baseline CALDERA under different quantization constraints.

4-Bit LR on Llama2. Table 2 reports results where the $\mathbf{L}$ and $\mathbf{R}$ are quantized to 4-Bit, with LPLR applied for iterative updates. Despite this additional optimization step, ODLRI consistently improves perplexity (WikiText-2, C4) and zero-shot accuracy across multiple tasks (e.g., PiQA, RTE) for most configurations. These results highlight that a principled initialization strategy enhances performance, even under aggressive quantization.

Importantly, the only modification from standard CALDERA is the $\mathbf{LR}$ initialization strategy, demonstrating that even in highly constrained quantization settings, ODLRI retains crucial distributional information that would otherwise can be lost. Additional results under more extreme compression, specifically at lower ranks ($r \leq 32$), can be found in Appendix C.2.

Model	Method	Rank	Avg Bits	PPL ↓	
				Wiki2	C4
7B	CALDERA	64	2.40	7.25	9.52
	+ODLRI	64	2.40	**7.17**	**9.41**
	CALDERA	128	2.80	6.84	8.95
	+ODLRI	128	2.80	**6.70**	**8.79**
	CALDERA	256	3.60	6.42	8.43
	+ODLRI	256	3.60	**6.18**	**8.23**
13B	CALDERA	64	2.32	5.93	**7.95**
	+ODLRI	64	2.32	**5.90**	7.96
	CALDERA	128	2.64	5.77	7.69
	+ODLRI	128	2.64	**5.64**	**7.54**
	CALDERA	256	3.28	5.48	7.32
	+ODLRI	256	3.28	**5.38**	**7.14**
Uncompressed (7B)		—	16	5.12	6.63
Uncompressed (13B)		—	16	4.57	6.05

Table 3: Zero-Shot Perplexity (↓) of ODLRI with CALDERA for Llama2 models on WikiText-2 and C4. $\mathbf{Q}$ is 2-Bit quantized, and $\mathbf{L}, \mathbf{R}$ are 16-Bit. Lower values are **bolded**. Additional zero-shot accuracy results are provided in Appendix C.1.

16-Bit LR on Llama2. We further evaluate the models in a 16-Bit $\mathbf{LR}$ setting, where the $\mathbf{LR}$ component is left unquantized (Table 3).

Without the need for low-bit refinement steps on $\mathbf{LR}$ component, this configuration allows for a direct assessment of ODLRI's effectiveness. It en-

ables $\mathbf{L}_0, \mathbf{R}_0$ to fully capture the benefits of outlier-driven initialization, preserving fine-grained weight structures that are critical for performance under extreme low-bit settings. As expected, perplexity and zero-shot accuracy improve compared to the more constrained 4-Bit LR scenario, providing clearer evidence of ODLRI's effectiveness without the confounding effects of additional $\mathbf{LR}$ quantization. As in the 4-Bit $\mathbf{LR}$ setting, ODLRI achieves more pronounced gains, particularly at higher ranks. The corresponding zero-shot accuracy measurements for 16-Bit LR setting are provided in Appendix C.1.

These results demonstrate that the effectiveness of ODLRI holds across both aggressive 4-Bit $\mathbf{LR}$ settings and relaxed 16-Bit $\mathbf{LR}$ conditions without additional refinement, highlighting its robustness across diverse compression regimes.

4-Bit LR on Llama3-8B and Mistral-7B. To assess the generalizability of ODLRI beyond Llama2 models, we evaluate its performance on Llama3-8B and Mistral-7B. Table 4 presents results for these models with 4-Bit quantization applied to the $\mathbf{LR}$ component. Under a range of rank configurations, ODLRI consistently improves over original CALDERA by more effectively capturing salient weight directions, resulting in lower perplexity on both WikiText-2 and C4. These results demonstrate that the effectiveness of ODLRI generalizes beyond Llama2 models. We also provide results on non-LLaMA models and alternative quantizers in Appendix C.3.

Rank	Method	Llama3-8B		Mistral-7B	
		Wiki2 ↓	C4 ↓	Wiki2 ↓	C4 ↓
64	CALDERA	10.58	11.35	**6.37**	7.11
	+ODLRI	**10.35**	**11.15**	**6.37**	**7.10**
128	CALDERA	9.41	**10.21**	6.11	6.89
	+ODLRI	**9.35**	10.32	**6.08**	**6.86**
256	CALDERA	8.70	9.77	5.77	6.59
	+ODLRI	**8.12**	**9.33**	**5.69**	**6.53**

Table 4: Zero-Shot Perplexity (↓) of ODLRI with CALDERA for Llama3-8B and Mistral-7B on WikiText-2 and C4. $\mathbf{Q}$ is 2-Bit quantized, while $\mathbf{L}, \mathbf{R}$ are 4-Bit quantized. Lower values are **bolded**.

4.4 Number of Outlier Columns (k)

In ODLRI, we determine salient weight components by explicitly targeting activation outliers, selecting top-k activations corresponding to outlier-sensitive weights. Instead of following the rank-based selection that picks the top-r components based solely on low-rank dimension r, we set $k < r$ to intensively focus on outliers.

ODLRI	$\mathbf{L}, \mathbf{R}$ 16-Bit		$\mathbf{L}, \mathbf{R}$ 4-Bit	
	Wiki2 ↓	C4 ↓	Wiki2 ↓	C4 ↓
$\mathbf{H}_o$ $(k = r)$	6.38	8.43	6.46	8.52
$\mathbf{H}_o$ $(k < r)$	**6.18**	**8.23**	**6.33**	**8.27**

Table 5: Comparison of OLDRI with various values of k, specifically $\mathbf{H_o}$ $(k = 256)$ and $\mathbf{H_o}$ $(k = 16)$ by perplexities (↓) of Llama2-7B on WikiText-2 and C4. $\mathbf{Q}$ is 2-Bit, and $\mathbf{L}$, $\mathbf{R}$ are either 16-Bit or quantized 4-Bit with a rank of 256. Lower perplexity values are **bolded**. Details regarding selection of k are provided in Appendix B.2.

To evaluate the impact of this selection, we measure perplexity (PPL) under different configurations. Our results show that choosing k based on activation outliers consistently outperforms selecting the top-r components through standard low-rank approximation. This improvement highlights the importance of leveraging activation statistics to guide low-rank approximation, ensuring that the most outlier-sensitive weights are efficiently modeled in $\mathbf{LR}$. These findings validate that ODLRI's targeted selection enhances low-rank approximation, leading to better quantization performance in extreme low-bit settings.

5 Conclusion

In this work, we investigated the role of initialization strategies in iterative joint optimization approaches for LLM compression. We introduce a unified framework that reformulates joint optimization algorithms through the lens of low-rank component initialization. Our analysis shows that the choice of initialization determines the entire trajectory of weight decomposition by assigning persistent roles to the components. Based on these insights, we proposed ODLRI, which assigns a distinct role to the low-rank component to capture activation-sensitive weights while using quantization for the remaining weights. Through extensive experiments across various LLM architectures, we demonstrated that incorporating ODLRI significantly improves model performance and compression stability. Consequently, our approach advances the practical implementation of efficient LLM compression and steers us toward optimal weight decomposition.

Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2024-00457882, National AI Research Lab Project), IITP grant funded by the Korean Government (MSIT) (No. RS-2020-II201361, Artificial Intelligence Graduate School Program (Yonsei University)), and K-CHIPS(Korea Collaborative & High-tech Initiative for Prospective Semiconductor Research) (RS-2024-00405946, 24052-15TC) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

Limitations

Our work focuses on weight-only quantization, optimizing the decomposition of model weights into quantized and low-rank components. While this approach enhances low-bit quantization performance, it does not address the quantization of activations or KV cache, which are critical for further improving inference efficiency. Joint weight and activation quantization introduces additional challenges, such as distributional shifts and increased sensitivity to outliers, requiring specialized calibration techniques. Similarly, KV cache quantization is essential for reducing memory overhead in long-context inference but remains outside the scope of this study.

Additionally, while we evaluate ODLRI within CALDERA, we believe that our approach can be applied to other joint $\mathbf{Q} + \mathbf{LR}$ optimization algorithms beyond CALDERA. Exploring how ODLRI integrates with different iterative quantization frameworks presents an interesting direction for future research.

Ethical Considerations

We focus on efficient compression of LLMs through quantization and low-rank decomposition. While these techniques improve computational efficiency and enable broader deployment, they also present ethical considerations.

First, compressed models may inherit and amplify biases present in the original model. Ensuring fairness and preventing unintended distortions due to quantization artifacts remain critical challenges.

Second, model compression enhances accessibility but may also facilitate misuse, including deployment in applications without adequate ethical oversight. Responsible usage and regulation are essential.

We emphasize the importance of ethical AI development and encourage continued evaluation of the societal impact of model compression.

References

Haoli Bai, Lu Hou, Lifeng Shang, Xin Jiang, Irwin King, and Michael R Lyu. 2022. Towards efficient post-training quantization of pre-trained language models. In *NeurIPS*.

Ron Banner, Yury Nahshan, and Daniel Soudry. 2019. Post training 4-bit quantization of convolutional networks for rapid-deployment. In *NeurIPS*.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. *TAC*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. 2023. Quip: 2-bit quantization of large language models with guarantees. In *NeurIPS*.

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*.

Bita Darvish Rouhani, Ritchie Zhao, Venmugil Elango, Rasoul Shafipour, Mathew Hall, Maral Mesmakhosroshahi, Ankit More, Levi Melnick, Maximilian Golub, Girish Varatkar, et al. 2023. With shared microexponents, a little shifting goes a long way. In *ISCA*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. In *NeurIPS*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023a. Qlora: efficient finetuning of quantized llms. In *NeurIPS*.

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. 2023b. Spqr: A sparse-quantized representation for near-lossless llm weight compression. *arXiv preprint arXiv:2306.03078*.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael Mahoney, and Kurt Keutzer. 2019. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *ICCV*.

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

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. OPTQ: Accurate quantization for generative pre-trained transformers. In *ICLR*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024a. A framework for few-shot language model evaluation.

Shangqian Gao, Ting Hua, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. 2024b. Adaptive rank selections for low-rank approximation of language models. In *NAACL-HLT*.

G.H. Golub, Alan Hoffman, and G.W. Stewart. 1987. A generalization of the eckart-young-mirsky matrix approximation theorem. *Linear Algebra and its Applications*.

Han Guo, Philip Greengard, Eric Xing, and Yoon Kim. 2024. Lq-lora: Low-rank plus quantized matrix decomposition for efficient language model finetuning. In *ICLR*.

Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Xianglong Liu, Luca Benini, Michele Magno, and Xiaojuan Qi. 2024. Slim-llm: Salience-driven mixed-precision quantization for large language models. *arXiv preprint arXiv:2405.14917*.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *CVPR*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

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

Changhun Lee, Jungyu Jin, Taesu Kim, Hyungjun Kim, and Eunhyeok Park. 2024. Owq: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models. In *AAAI*.

Liang Li, Qingyuan Li, Bo Zhang, and Xiangxiang Chu. 2024a. Norm tweaking: High-performance low-bit quantization of large language models. In *AAAI*.

Yixiao Li, Yifan Yu, Chen Liang, Nikos Karampatziakis, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2024b. Loftq: LoRA-fine-tuning-aware quantization for large language models. In *ICLR*.

Yixiao Li, Yifan Yu, Qingru Zhang, Chen Liang, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2023. Losparse: structured compression of large language models based on low-rank and sparse approximation. In *ICML*.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. 2021. Brecq: Pushing the limit of post-training quantization by block reconstruction. In *ICLR*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. In *MLSys*.

Shih-Yang Liu, Huck Yang, Chien-Yi Wang, Nai Chit Fung, Hongxu Yin, Charbel Sakr, Saurav Muralidharan, Kwang-Ting Cheng, Jan Kautz, Yu-Chiang Frank Wang, et al. 2024. Eora: Training-free compensation for compressed llm with eigenspace low-rank approximation. *arXiv preprint arXiv:2410.21271*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *ICLR*.

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. 2020. Up or down? adaptive rounding for post-training quantization. In *ICML*.

Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart Van Baalen, and Tijmen Blankevoort. 2021. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*.

Antonio Polino, Razvan Pascanu, and Dan Alistarh. 2018. Model compression via distillation and quantization. In *ICLR*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

RelaxML. 2025. Hessians-llama-2 models (7b, 13b, 70b) - 6144. Accessed: 2025-02-09. Models available at: https://huggingface.co/relaxml.

Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, et al. 2024. Gemma 2: Improving open language models at a practical size. In *CoRR*.

Rajarshi Saha, Naomi Sagan, Varun Srivastava, Andrea Goldsmith, and Mert Pilanci. 2024. Compressing large language models using low rank and low precision decomposition. In *NeurIPS*.

Rajarshi Saha, Varun Srivastava, and Mert Pilanci. 2023. Matrix compression via randomized low rank and low precision factorization. In *NeurIPS*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *AAAI*.

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 *ICLR*.

Pierre Stock, Armand Joulin, Rémi Gribonval, Benjamin Graham, and Hervé Jégou. 2020. And the bit goes down: Revisiting the quantization of neural networks. In *ICLR*.

Chaofan Tao, Lu Hou, Wei Zhang, Lifeng Shang, Xin Jiang, Qun Liu, Ping Luo, and Ngai Wong. 2022. Compression of generative pre-trained language models via quantization. In *ACL*.

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*.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. 2024. Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks. In *ICML*.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. 2025. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *ICLR*.

Maurice Weber, Daniel Y Fu, Quentin Gregory Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Re, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. In *NeurIPS Datasets and Benchmarks Track*.

Hao Wu, Patrick Judd, Xiaojie Zhang, Mikhail Isaev, and Paulius Micikevicius. 2020. Integer quantization for deep learning inference: Principles and empirical evaluation. *arXiv preprint arXiv:2004.09602*.

Jaewoo Yang, Hayun Kim, and Younghoon Kim. 2024. Mitigating quantization errors due to activation spikes in glu-based llms. *arXiv preprint arXiv:2405.14428*.

Zhewei Yao, Xiaoxia Wu, Cheng Li, Stephen Youn, and Yuxiong He. 2024. Exploring post-training quantization in llms from comprehensive study to low rank compensation. In *AAAI*.

Mengxia Yu, De Wang, Qi Shan, and Alvin Wan. 2024. The super weight in large language models. *arXiv preprint arXiv:2411.07191*.

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

Cheng Zhang, Jianyi Cheng, George Anthony Constantinides, and Yiren Zhao. 2024. Lqer: Low-rank quantization error reconstruction for llms. In *ICML*.

Cheng Zhang, Jeffrey T. H. Wong, Can Xiao, George Anthony Constantinides, and Yiren Zhao. 2025. QERA: an analytical framework for quantization error reconstruction. In *ICLR*.

A Additional Experimental Setup

Calibration Data and Hessian Computation. We use precomputed Hessians from RelaxML (2025) for Llama2 models. For Llama3-8B and Mistral-7B, Hessians are computed directly to ensure alignment with model-specific architectures. Hessians are computed using 256 randomly sampled examples from the RedPajama dataset (Weber et al., 2024). Context lengths are set to 4096 tokens for Llama2 and 8192 tokens for Llama3, maintaining consistency with each model's native configuration.

Hardware Environment. Experiments were conducted on both consumer- and enterprise-grade GPUs. The primary pipeline was executed on NVIDIA GeForce RTX 3090, RTX 4090, and RTX A6000 GPUs, while the quantization and evaluation of the Llama2-70B model were performed on NVIDIA L40s GPU clusters due to its large size.

Computation time for quantization and evaluation varied depending on the model size and hardware configuration. For Llama2-7B and Mistral-7B, quantization was completed in 4 GPU hours on a cluster of six GPUs. For Llama2-13B, quantization required 8 GPU hours on a six-GPU cluster. Besides, for Llama3-8B, quantization was achieved in 5 GPU hours on a cluster of six GPUs. For Llama2-70B, quantization was conducted using four NVIDIA L40S GPUs, completing within 48 hours. In the absence of parallel processing, the estimated runtime would have increased proportionally to the number of GPUs used. Evaluation phase was carried out on the same GPUs utilized for quantization. Perplexity measurement required approximately 0.5 GPU hours, while zero-shot evaluation consumed around 1.5 GPU hours. Since pretrained models were used, no additional training time was required.

All experiments were repeated using two different random seeds, with the exception of Llama2-70B, for which only one seed was used due to resource constraints. Overall, the quantization and evaluation processes accumulated approximately 4000 GPU hours in total.

CALDERA's Default Configuration. In our experiments, the CALDERA model was configured with the following settings: `hadamard_transform` was set to `true`, `outer_iter` to `15`, `inner_iter` to `10`, `rand_svd` to `false`, `Q_hessian_downdate` to `false`, and `update_order` to `Q LR`.

Summary of Model and Dataset. We present a summary of the models and datasets employed in this paper. The detailed specifications, including sources, access methods, and licensing terms, are summarized in Table 6. Also, the datasets and their corresponding metadata are detailed in Table 7.

B Outlier-Aware Initialization Detail

B.1 ODLRI Method in Detail

To process outlier-sensitive weights by low-rank approximation, we decompose our optimization objective

$$\mathbf{WX} = \mathbf{W}(\mathbf{X}_\mathrm{o} + \mathbf{X}_\mathrm{r}) = \mathbf{WX}_\mathrm{o} + \mathbf{WX}_\mathrm{r}$$

into two distinct components. Here, $\mathbf{X}_\mathrm{o}$ represents the channels containing outliers (with other channels set to zero), while $\mathbf{X}_\mathrm{r}$ indicates the remaining channels (with outlier channels zeroed). Since the non-zero entries of $\mathbf{X}_\mathrm{o}$ and $\mathbf{X}_\mathrm{r}$ are mutually exclusive along the channel dimension, both $\mathbf{X}_\mathrm{o}$ and $\mathbf{X}_\mathrm{r}$ retain the original matrix dimensions.

Standard activation-aware low-rank factorization, which solves

$$\arg\min_{\mathbf{L},\mathbf{R}} \|(\mathbf{W} - \mathbf{LR})\mathbf{X}\|_\mathrm{F}^2.$$

Using the empirical second-moment matrix $\mathbf{H} = \mathbf{XX}^\top$, this objective is equivalent to:

$$\arg\min_{\mathbf{L},\mathbf{R}} \|(\mathbf{W} - \mathbf{LR})\mathbf{H}(\mathbf{W} - \mathbf{LR})^\top\|.$$

However, this formulation treats all activations equally, leading to a low-rank approximation that does not explicitly focus on outliers, which are often the primary bottleneck in quantization.

To ensure that the low-rank component captures only the most challenging weight structures, we define a restricted activation covariance matrix $\mathbf{H}_\mathrm{o}$, a submatrix of $\mathbf{H}$ that prioritizes outlier-sensitive channels:

$$(\mathbf{H}_o)_{ij} = \begin{cases} \mathbf{H}_{ij}, & \text{if } i,j \in \mathcal{I}, \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where the subset of indices $\mathcal{I} \subset \{1,\dots,d\}$ corresponds to the top-k ($k < r$) channels with the highest diagonal values of $\mathbf{H}$. These channels correspond to the most dominant activation patterns, which often align with weight outliers that distort quantization performance.

Model	Source	Accessed via	License
Llama2-7B	(Touvron et al., 2023)	Link	Llama 2 Community License
Llama2-13B	(Touvron et al., 2023)	Link	Llama 2 Community License
Llama2-70B	(Touvron et al., 2023)	Link	Llama 2 Community License
Llama3-8B	(Dubey et al., 2024)	Link	Meta Llama 3 Community License
Mistral-7Bv0.1	(Jiang et al., 2023)	Link	Apache license 2.0

Table 6: Summary of models used in this paper.

Dataset	Source	Accessed via	License
RedPajama-Data-1T-Sample	(Weber et al., 2024)	Link	Apache License 2.0
Hessians-Llama-2-7b-6144	(RelaxML, 2025)	Link	-
Hessians-Llama-2-13b-6144	(RelaxML, 2025)	Link	-
Hessians-Llama-2-70b-6144	(RelaxML, 2025)	Link	-
Wikitext-2-raw-v1	(Merity et al., 2017)	Link	CC-BY-SA-3.0
C4	(Raffel et al., 2020)	Link	ODC-BY
lm-eval-harness	(Gao et al., 2024a)	Link	MIT License
Winogrande	(Sakaguchi et al., 2020)	Link	Apache License 2.0
RTE	(Bentivogli et al., 2009)	Link	Apache License 2.0
PiQA	(Bisk et al., 2020)	Link	Apache License 2.0
ARC Easy	(Clark et al., 2018)	Link	CC-BY-SA-4.0
ARC Challenge	(Clark et al., 2018)	Link	CC-BY-SA-4.0

Table 7: Summary of datasets used in this paper.

We then solve the outlier-aware optimization problem:

$$\mathbf{L}_0, \mathbf{R}_0 = \arg\min_{\mathbf{L},\mathbf{R}} \|(\mathbf{W} - \mathbf{LR})\mathbf{H}_\text{o}(\mathbf{W} - \mathbf{LR})^\top\|.$$

This is achieved by applying a Cholesky decomposition to the selected Hessian submatrix $\mathbf{H}_\text{o}$, which encodes the activation covariance information of high-variance channels:

$$\mathbf{H}_\text{o} = \mathbf{S}_\text{o}\mathbf{S}_\text{o}^\top,$$

where $\mathbf{S}_\text{o}$ is a lower triangular matrix.

Unlike SVD-LLM (Wang et al., 2025), which applies data whitening to the entire Hessian matrix $\mathbf{H}$, our transformation performs selective whitening on the outlier subset $\mathbf{H}_\text{o}$ corresponding to $\mathbf{X}_\text{o}$. This selective whitening improves the numerical conditioning for the subsequent SVD while ensuring that the low-rank component effectively captures the salient weight information.

Once the whitening transformation is applied, we perform SVD on the transformed weight matrix $\mathbf{WS}_\text{o}$, truncating the decomposition to rank r to obtain the outlier-focused low-rank components:

$$\text{SVD}(\mathbf{WS}_\text{o}) = \mathbf{U}_{:,:r}\mathbf{\Sigma}_{:r,:r}\mathbf{V}_{:r,:}^\top.$$

Here, the Python-style slicing notation $\mathbf{U}_{:,:r}$ indicates that we select all rows and the first r columns of $\mathbf{U}$. Similarly, $\mathbf{\Sigma}_{:r,:r}$ denotes the upper-left $r \times r$ block of $\mathbf{\Sigma}$, and $\mathbf{V}_{:r,:}$ selects the first r rows of $\mathbf{V}$.

To ensure that the decomposition remains truncation-aware, we initialize the low-rank and quantization components as follows:

$$\begin{aligned}\mathbf{L}_0 &= \mathbf{U}_{:,:r}\sqrt{\mathbf{\Sigma}_{:r,:r}} \\ \mathbf{R}_0 &= \sqrt{\mathbf{\Sigma}_{:r,:r}}\mathbf{V}_{:r,:}^\top\mathbf{S}_\text{o}^{-1}.\end{aligned}$$

By ensuring that the residual weight matrix $\mathbf{W} - \mathbf{L}_0\mathbf{R}_0$ has been preconditioned to remove outlier effects, we significantly reduce quantization error compared to conventional activation-aware quantization approaches. This allows the quantized component $\mathbf{Q}$ to operate on a more uniform weight distribution, improving numerical stability:

$$\mathbf{Q}_1 = \text{Quantize}(\mathbf{W} - \mathbf{L}_0\mathbf{R}_0).$$

B.2 Rank-Dependent Outlier Selection of k

In our experiments, we denote the rank of the low-rank components $\mathbf{L}$ and $\mathbf{R}$ as r, and we set the number of outlier-sensitive columns k in proportion to this rank. Specifically, we define:

$$k = p \times n,$$

where n is the dimension of the calibration Hessian $\mathbf{H} \in \mathbb{R}^{n \times n}$ and p is a rank-dependent percentage.

We adopt the following settings:

- For $r = 64$: $p = 0.1\%$
- For $r = 128$: $p = 0.2\%$
- For $r = 256$: $p = 0.4\%$

For example, consider the key projection matrix of Llama2-7B. Its corresponding Hessian matrix has a shape of 4096×4096. When we set the rank $r = 256$, we use $p = 0.4\%$. Thus, the number of outlier-sensitive columns is computed as:

$$k = p \times n = 0.4\% \times 4096 \approx 16.$$

This means that in this example, 16 outlier-sensitive columns are selected for the key projection.

B.3 ODLRI's Impact on Salient Weights

Hessian	$\frac{\|\mathbf{LRX}_\text{o}\|}{\|\mathbf{WX}_\text{o}\|}$	$\frac{\|\mathbf{E}_{LR}\mathbf{X}_\text{o}\|}{\|\mathbf{WX}_\text{o}\|}$	$\frac{\|\mathbf{LRX}_r\|}{\|\mathbf{WX}_r\|}$	$\frac{\|\mathbf{E}_{LR}\mathbf{X}_r\|}{\|\mathbf{WX}_r\|}$
$\mathbf{H}$	0.997	0.073	0.920	0.392
$\mathbf{H}_o$	0.999	0.001	0.903	0.430

Table 8: Effect of Hessian selections in ODLRI. Results are shown for Layer 10 Key Projection matrix of Llama2-7B when using ODLRI initialization. We compare normalized norm of $\|\mathbf{X}_o\|$ and $\|\mathbf{X}_r\|$ for $\mathbf{LR}$ and $\mathbf{E}_{LR}$ when using ODLRI initialization. L2-norm is denoted by $\|\cdot\|$. And $\mathbf{E}_{LR} = \mathbf{W} - \mathbf{LR}$.

This section validates that ODLRI effectively captures and preserves salient weights by explicitly targeting outlier activations in $\mathbf{X}$. Unlike standard low-rank approximations that distribute capacity across all activations, ODLRI selectively focuses on outlier activations, ensuring that the low-rank component is structured to represent the most outlier-sensitive weights before quantization.

To demonstrate this, we compare two hessian formulations for guiding the low-rank approximation. The first uses the full activation matrix $\mathbf{H} = \mathbf{X}\mathbf{X}^\top$, while the second employs only the top-k outlier activations, $\mathbf{X}_\text{o}$, forming the restricted hessian matrix $\mathbf{H}_\text{o} = \mathbf{X}_\text{o}\mathbf{X}_\text{o}^\top$. The effectiveness of each approach is assessed by evaluating the low-rank representation $\mathbf{LR}$ through the activation norm $\|\mathbf{LRX}_\text{o}\|$ compared to $\|\mathbf{WX}_\text{o}\|$.

Table 8 presents that using $\mathbf{H}_\text{o}$ yields a significantly closer approximation of $\mathbf{WX}_\text{o}$ than using $\mathbf{H}$, confirming that ODLRI's outlier-driven initialization better preserves salient weights. Moreover, the approximation for non-salient weights ($\mathbf{X}_\text{r}$) remains stable, demonstrating that prioritizing activation outliers does not degrade overall representation.

C Additional Results

C.1 16-Bit LR on Llama2

Table 9 presents the results of zero-shot accuracy under the setting of 2-Bit $\mathbf{Q}$ and 16-Bit $\mathbf{LR}$. Overall, the results show that the method incorporating ODLRI outperforms the baseline CALDERA approach.

C.2 Lower Rank, Extreme Compression on Llama2

To further evaluate the robustness of our method under more aggressive compression settings, we conduct additional experiments at significantly lower ranks, specifically $r = 16$ and $r = 32$. These settings simulate extreme compression scenarios, where the capacity of the low-rank residual becomes severely constrained.

We evaluate on the Llama2-7B model using the same setup as our main experiments: $\mathbf{Q}$ is quantized to 2 bits, and $\mathbf{L}, \mathbf{R}$ to 4 bits. Table 10 reports perplexity and accuracy across a range of zero-shot evaluation benchmarks.

Despite the reduced rank and further limited representational capacity, our proposed ODLRI initialization continues to yield improvements over CALDERA. These findings confirm that ODLRI remains effective under severe rank constraints, preserving performance.

C.3 ODLRI on non-Llama Models and Alternative Quantizers.

Abnormal outliers have been observed in Llama-style LLMs (Yang et al., 2024; Yu et al., 2024), including the Llama family and Mistral, possibly due to their model architecture. To assess whether our outlier-aware method remains effective beyond this setting, we evaluate the robustness of ODLRI in two distinct scenarios: (1) non-Llama architectures

Model	Method	Rank	Zero-Shot Accuracy ↑				
			Wino	RTE	PiQA	ArcE	ArcC
7B	CALDERA	64	63.73	**55.59**	**73.4**	61.93	31.27
	+ODLRI	64	**65.75**	55.23	72.91	**64.44**	**31.74**
	CALDERA	128	**63.97**	**59.03**	73.53	64.58	32.94
	+ODLRI	128	63.93	**59.03**	**73.64**	**64.86**	**33.75**
	CALDERA	256	**66.1**	60.47	**74.45**	64.62	34.00
	+ODLRI	256	64.57	**61.46**	65.12	**71.58**	**36.31**
	QuIP#	0	61.7	57.8	69.6	61.2	29.9
13B	CALDERA	64	67.36	58.84	75.07	68.11	**37.58**
	+ODLRI	64	**69.85**	**67.51**	**75.41**	**69.91**	37.37
	CALDERA	128	**69.85**	64.98	75.9	**70.75**	38.82
	+ODLRI	128	68.98	**65.83**	**76.14**	69.59	**39.93**
	CALDERA	256	67.12	59.02	76.24	70.28	39.08
	+ODLRI	256	**70.40**	**67.15**	**76.55**	**71.63**	**41.04**
	QuIP#	0	63.6	54.5	74.2	68.7	36.2
Uncompressed (7B)		—	67.3	63.2	78.5	69.3	40.0
Uncompressed (13B)		—	69.5	61.7	78.8	73.2	45.6

Table 9: Comparison of our method with CALDERA by zero-shot accuracies (↑) of Llama2 models. $\mathbf{Q}$ is quantized 2-Bit and $\mathbf{L}, \mathbf{R}$ are 16-Bit. Higher accuracy values are are **bolded**.

Rank	Method	Avg Bits	PPL ↓		Zero-Shot Accuracy ↑				
			Wiki2	C4	Wino	RTE	PiQA	ArcE	ArcC
16	CALDERA	2.025	7.88	10.16	**62.64**	51.26	72.09	**60.63**	**31.48**
	+ODLRI	2.025	**7.79**	**10.02**	61.8	**59.57**	**72.36**	59.43	30.46
32	CALDERA	2.05	7.85	10.18	62.9	56.31	71.7	59.00	29.52
	+ODLRI	2.05	**7.47**	**9.75**	**65.03**	**62.45**	**72.52**	**62.28**	**31.65**

Table 10: Comparison of our method with CALDERA on zero-shot perplexities (↓) and accuracies (↑) of Llama2 models on extreme low bit setting. $\mathbf{Q}$ is quantized 2-Bit and $\mathbf{L}, \mathbf{R}$ are quantized 4-Bit. Lower perplexity and higher accuracy values are **bolded**.

Method	Llama2-7B		Gemma2-2B	
	$r = 32$	$r = 64$	$r = 32$	$r = 64$
FP16	8.71		13.08	
MXINT-base	10.62	10.38	18.72	17.65
+ODLRI	**10.57**	**10.26**	**18.52**	**17.17**

Table 11: Comparison of OLDRI with MXINT-base on zero-shot perplexities (↓) of Llama2-7B and Gemma2-2B on low bit setting. $\mathbf{Q}$ is quantized 3-Bit and $\mathbf{L}$,$\mathbf{R}$ are 16-Bit. Lower perplexity values are **bolded**.

and (2) alternative quantization methods. Specifically, we conduct experiments on both Gemma2-2B (Rivière et al., 2024) and Llama2-7B using the MXINT (Darvish Rouhani et al., 2023) quantizer. We replace QuIP# (Tseng et al., 2024) with MXINT (3-bit, block size 32) for both models.

We define MXINT-base as a baseline that follows the same quantize-then-approximate structure as CALDERA: it first applies MXINT quantization and then performs low-rank approximation using the same activation-aware SVD used in CALDERA, without any outlier-driven refinement.

Gemma2-2B serves to test whether ODLRI generalizes to architectures outside the Llama family. In contrast, Llama2-7B is included to isolate and examine the impact of changing the quantizer while keeping the model architecture fixed. For both models, we adopt the same activation-aware low-rank approximation method as CALDERA and perform 15 outer iterations alternating between quantization and low-rank refinement.

Perplexity is measured using `lm-eval-harness`, which may yield different values from those in the main text. Hence, we also report FP16 perplexities for reference. We note that the $\mathbf{LR}$ component is kept in 16-Bit precision for all experiments. As shown in Table 11, ODLRI consistently reduces perplexity compared to MXINT-base. These findings confirm that ODLRI remains effective across both non-Llama models and alternative quantiza-

tion schemes.

C.4 Effect of LR Initialization in CALDERA

Weight Type	Iteration	Initialization Method			
		0		LRApprox(W)	
		$\|\mathbf{QX}\|$	$\|\mathbf{LRX}\|$	$\|\mathbf{QX}\|$	$\|\mathbf{LRX}\|$
Key Proj.	First	0.999	0.014	0.158	0.915
	Last	0.961	0.073	0.401	0.664
Query Proj.	First	0.999	0.014	0.148	0.924
	Last	0.956	0.073	0.408	0.657
Value Proj.	First	0.993	0.072	0.378	0.885
	Last	0.970	0.264	0.622	0.676
O Proj.	First	0.993	0.038	0.373	0.886
	Last	0.958	0.149	0.495	0.734
Up Proj.	First	0.978	0.064	0.605	0.766
	Last	0.966	0.198	0.666	0.684
Gate Proj.	First	0.986	0.048	0.443	0.824
	Last	0.951	0.165	0.584	0.626
Down Proj.	First	1.000	0.013	0.071	0.976
	Last	0.996	0.049	0.181	0.869

Table 12: Effect of $\mathbf{LR}$ initialization strategies in CALDERA. Results are shown for Layer 1's weight matrices from Llama2-7B over 15 iterations. We compare $\|\mathbf{QX}\|$ and $\|\mathbf{LRX}\|$, both normalized by $\|\mathbf{WX}\|$ (i.e., $\|\mathbf{QX}\|/\|\mathbf{WX}\|$) at first and last iterations. L2-norm is denoted by $\|\cdot\|$.

Weight Type	Iteration	Initialization Method			
		0		LRApprox(W)	
		$\|\mathbf{QX}\|$	$\|\mathbf{LRX}\|$	$\|\mathbf{QX}\|$	$\|\mathbf{LRX}\|$
Key Proj.	First	0.995	0.040	0.278	0.869
	Last	0.960	0.106	0.552	0.575
Query Proj.	First	0.992	0.049	0.324	0.847
	Last	0.960	0.127	0.549	0.606
Value Proj.	First	0.976	0.094	0.570	0.792
	Last	0.966	0.220	0.652	0.722
O Proj.	First	0.978	0.102	0.644	0.733
	Last	0.970	0.265	0.725	0.680
Up Proj.	First	0.978	0.077	0.621	0.740
	Last	0.978	0.182	0.676	0.686
Gate Proj.	First	0.987	0.057	0.475	0.786
	Last	0.975	0.139	0.606	0.608
Down Proj.	First	0.971	0.090	0.769	0.591
	Last	0.971	0.208	0.814	0.552

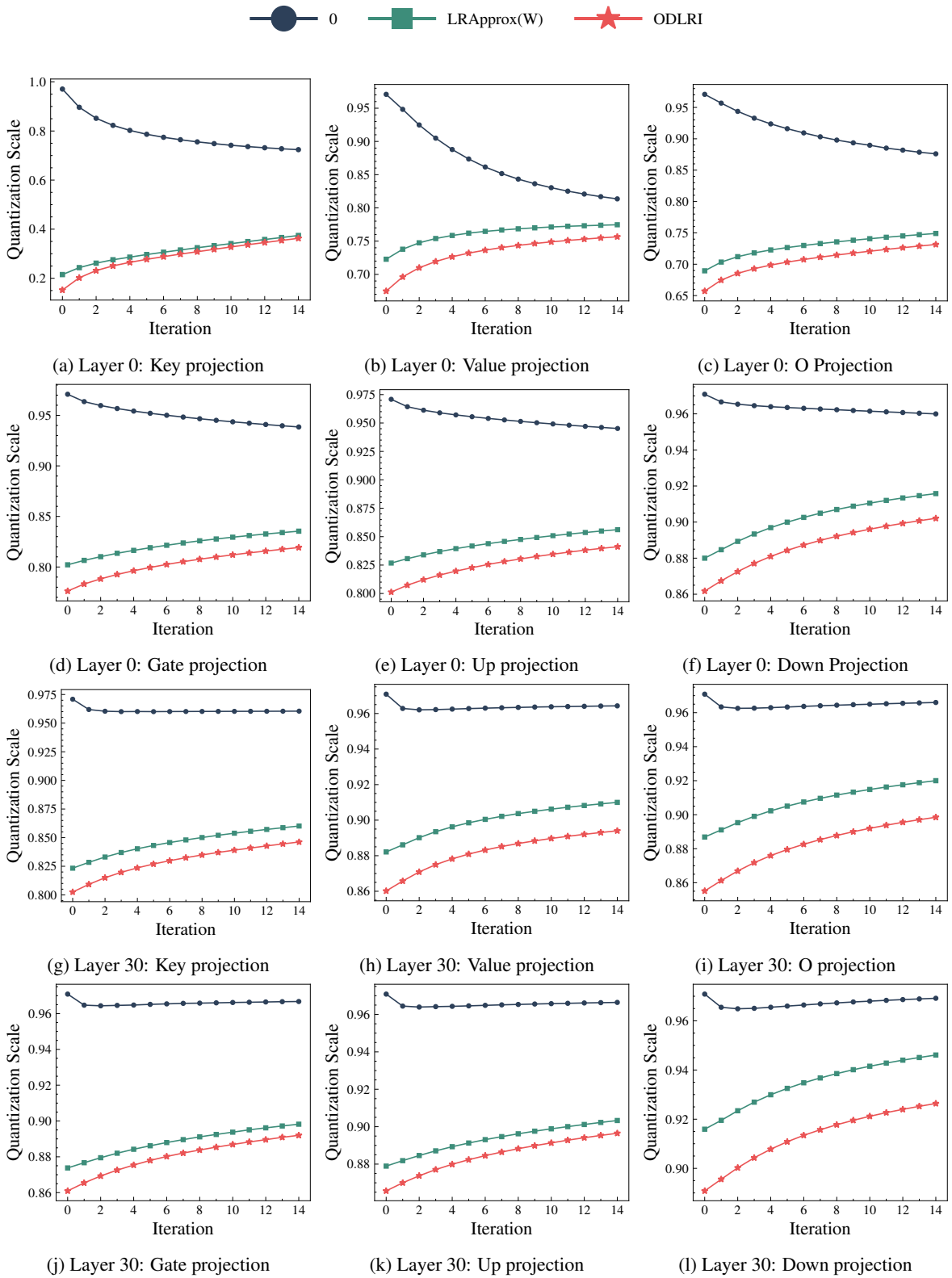
Table 13: Effect of $\mathbf{LR}$ initialization strategies in CALDERA. Results are shown for Layer 10's weight matrices from Llama2-7B over 15 iterations. We compare $\|\mathbf{QX}\|$ and $\|\mathbf{LRX}\|$, both normalized by $\|\mathbf{WX}\|$ (i.e., $\|\mathbf{QX}\|/\|\mathbf{WX}\|$) at first and last iterations. L2-norm is denoted by $\|\cdot\|$.

Table 12 and Table 13 present how different $\mathbf{LR}$ initialization strategies affect the final weight distributions in Layer 1 and Layer 10, respectively. In both layers, we observe that varying the initialization leads to significantly different outcomes across all weights. This finding reveals that joint optimization outcomes are highly sensitive to initialization choices, which ultimately determine whether quantization or matrix factorization dominates the final representation.

C.5 Effect of LR Initialization on Joint Optimization for Q+LR

Figure 4 examines the quantization scale across various layers, demonstrating that our method, ODLRI, effectively maintains optimal scale control throughout the network. The results indicate that ODLRI consistently outperforms baseline approaches in ensuring robust and stable quantization across diverse layers.

Similarly, Figure 5 presents the activation-aware error measured across multiple layers, confirming that ODLRI reliably minimizes activation-aware error in the preservation of intermediate activations. These findings underscore the efficacy of our approach in reducing activation-aware error, thereby enhancing overall model performance compared to baseline methods.



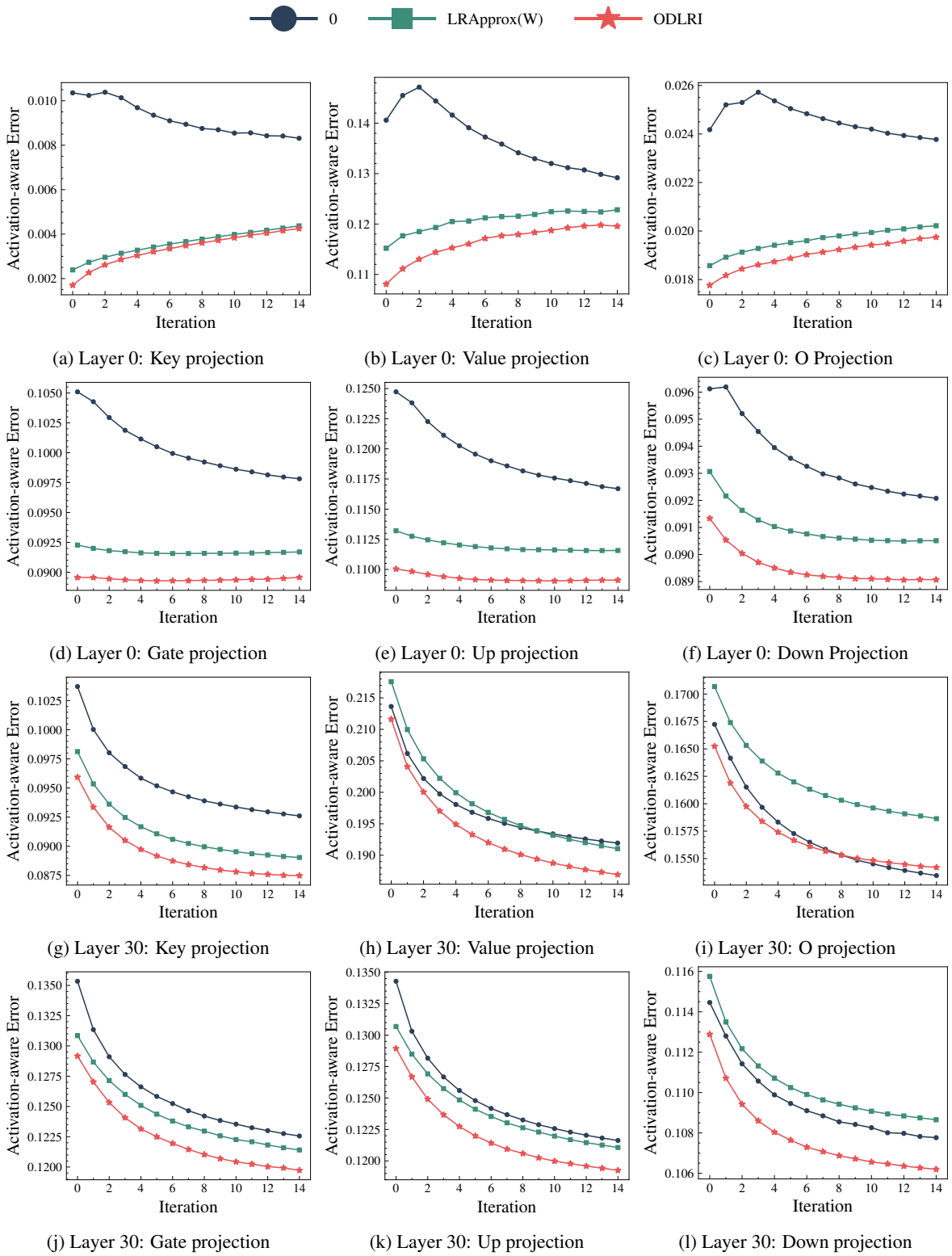
(a) Layer 0: Key projection (b) Layer 0: Value projection (c) Layer 0: O Projection

(d) Layer 0: Gate projection (e) Layer 0: Up projection (f) Layer 0: Down Projection

(g) Layer 30: Key projection (h) Layer 30: Value projection (i) Layer 30: O projection

(j) Layer 30: Gate projection (k) Layer 30: Up projection (l) Layer 30: Down projection

Figure 4: **Quantization Scale across different initialization strategies.** We present the quantization scale over 15 iterations, where both $\mathbf{L}$ and $\mathbf{R}$ are quantized to 4-Bit at rank 256. Subplots display results for the Key, Value, O, Gate, Up, and Down projection layers in Layer 0 and Layer 30 of Llama2-7B. ODLRI (red stars) consistently achieves the lowest quantization scale, highlighting its effectiveness in low-bit quantization.



(a) Layer 0: Key projection (b) Layer 0: Value projection (c) Layer 0: O Projection

(d) Layer 0: Gate projection (e) Layer 0: Up projection (f) Layer 0: Down Projection

(g) Layer 30: Key projection (h) Layer 30: Value projection (i) Layer 30: O projection

(j) Layer 30: Gate projection (k) Layer 30: Up projection (l) Layer 30: Down projection

Figure 5: **Activation-aware Error across different initialization strategies.** We present the normalized activation-aware error $\|(\mathbf{W}-\mathbf{Q}-\mathbf{LR})\mathbf{X}\|_{\mathrm{F}}^2/\|\mathbf{WX}\|_{\mathrm{F}}^2$ over 15 iterations, where both $\mathbf{L}$ and $\mathbf{R}$ are quantized to 4-Bit at rank 256. Lower values indicate better preservation of activation information after quantization. Subplots display results for the Key, Value, O, Gate, Up, and Down projection layers in Layer 0 and Layer 30 of Llama2-7B. ODLRI (red stars) consistently shows the lowest error, demonstrating its effectiveness in reducing quantization error.