# LRSR: HARDWARE-FRIENDLY APPROXIMATION OF PER-ELEMENT QUANTIZATION VIA LOW-RANK SCALE RECONSTRUCTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Post-training quantization is a crucial component for deploying Large Language Models (LLMs) in resource-constrained environments. While fine-grained methods (e.g., per-group) offer high accuracy, they heavily rely on hardware-specific implementations and suffer from memory-bound computations, limiting their applicability on standard matrix multiplication accelerators. Conversely, hardware-friendly quantizations (e.g., per-channel) experience severe accuracy degradation at low bit-widths. To address these limitations, we propose Low-Rank Scale Reconstruction (LRSR), a novel quantization method that approximates per-element scale matrices using an upper-bounding rank-1 decomposition, enabling mathematically correct reconstruction via general matrix multiplication (GEMM). Furthermore, we introduce several synergetic clusterization techniques to significantly boost accuracy and allow optimal hardware load balancing. Experimental evaluations on Qwen3 models demonstrate that at INT4, LRSR reduces KL-Divergence by up to 31% compared to the per-channel baseline, while utilizing 29$\times$ less metadata memory compared to per-group (g=128) quantization. This establishes a new Pareto frontier for near-zero overhead, hardware-friendly and calibration-free post-training quantizations.

## 1 INTRODUCTION

The deployment of neural networks and the reduction of their memory requirements remain among the most active research areas in deep learning. For Large Language Models (LLMs), post-training quantization has become a highly effective strategy to mitigate the immense memory bandwidth and computational costs associated with inference.

Recent advancements in quantization, such as vector quantization via learned codebooks, demonstrate exceptionally high information retention per memory unit. However, these methods share several critical drawbacks. The most significant limitation is their reliance on specialized hardware support, custom kernels, and complex calibration data. Consequently, they cannot be executed using standard General Matrix Multiplication (GEMM).

This limitation is a severe bottleneck in practical deployment. Modern inference frameworks are deeply optimized for standard matrix multiplication. Furthermore, many specialized computational devices - particularly Neural Processing Units (NPUs) - lack the flexibility to execute arbitrary custom operations, relying entirely on fixed multiply-accumulate (MAC) blocks designed strictly for standard GEMM. Therefore, research into "hardware-friendly" quantization methods is not only relevant but highly demanded by the industry.

To address this challenge, we propose Low-Rank Scale Reconstruction (LRSR), a calibration-free and GEMM-compatible quantization method. By redefining how fine-grained quantization scales are stored and applied, our approach achieves the accuracy benefits of high-resolution quantization without breaking standard matrix multiplication paradigms.

Our primary contributions are as follows:

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.

- We identify and mathematically exploit the factorability of low-rank scale factors out of the matrix multiplication contraction, introducing a GEMM-compatible quantization method with near-zero metadata overhead.
- We formulate a novel upper-bounding rank-1 decomposition algorithm that approximates scale matrices using the same methodology in its foundation as in per-channel and per-group granularities.
- We demonstrate that our approach establishes a new Pareto frontier in the calibration-free regime, significantly improving Top-1 token agreement and reducing KL-Divergence compared to standard quantization baselines.

## 2 BACKGROUND

### 2.1 SYMMETRIC QUANTIZATION AND GRANULARITY

Post-training quantization typically maps high-precision floating-point weights into a lower-precision integer format to accelerate inference. Let $\mathbf{W} \in \mathbb{R}^{N \times K}$ be the weight matrix of a linear layer, where $N$ is the number of output channels and $K$ is the number of input channels. In symmetric quantization, the quantized integer weights $\hat{\mathbf{W}}$ are obtained using a scale factor $s$:

$$\hat{\mathbf{W}} = \text{clip}\left(\left\lfloor \frac{\mathbf{W}}{s} \right\rceil, -q_{\max}, q_{\max}\right), \tag{1}$$

where $\lfloor \cdot \rceil$ denotes rounding to the nearest integer, and $q_{\max} = 2^{b-1} - 1$ represents the maximum representable value for a $b$-bit signed integer (e.g., $q_{\max} = 7$ for INT4).

The accuracy and computational profile of the quantized layer are heavily dictated by the granularity of the scale factor $s$. The most common granularities are per-tensor (one scale for the entire matrix), per-channel (one scale per output channel $N$), and per-group (one scale per block of $g$ elements along the input channel $K$).

### 2.2 HARDWARE-FRIENDLINESS AND THE NATURE OF GEMM

To understand why some quantization methods require specialized hardware support while others do not, it is necessary to examine the mechanics of general matrix multiplication. Consider the linear projection $\mathbf{Y} = \mathbf{W}_q\mathbf{A}$, where $\mathbf{W}_q$ represents the dequantized weights ($\mathbf{W}_q = s \cdot \hat{\mathbf{W}}$) and $\mathbf{A} \in \mathbb{R}^{K \times M}$ is the activation matrix. The computation for a single output element $\mathbf{Y}_{i,j}$ is defined as:

$$\mathbf{Y}_{i,j} = \sum_{k=1}^{K} (\mathbf{W}_q)_{i,k}\mathbf{A}_{k,j}. \tag{2}$$

In per-channel quantization, the scale factor $s_i$ depends only on the output channel index $i$. Consequently, it can be factored out of the summation:

$$\mathbf{Y}_{i,j} = \sum_{k=1}^{K} \left(s_i\hat{\mathbf{W}}_{i,k}\right)\mathbf{A}_{k,j} = s_i\left(\sum_{k=1}^{K} \hat{\mathbf{W}}_{i,k}\mathbf{A}_{k,j}\right). \tag{3}$$

This mathematical property allows the core operation to be executed as a single standard highly optimized GEMM ($\hat{\mathbf{W}}\mathbf{A}$), followed by a lightweight floating-point vector multiplication to apply $s_i$. This keeps all scale-dependent work outside the contraction, preserving full compatibility with standard MAC blocks.

In contrast, per-group quantization assigns a scale factor $s_{i,g(k)}$ that depends on the reduction axis index $k$. The equation becomes:

$$\mathbf{Y}_{i,j} = \sum_{k=1}^{K} s_{i,g(k)}\hat{\mathbf{W}}_{i,k}\mathbf{A}_{k,j}. \tag{4}$$

Because the scale varies along the summation index $k$, it cannot be factored out, and the dequantization multiplication must be performed element-by-element inside the dot product. This forces the hardware to continuously load scale factors from memory during the computation, which requires custom kernels that are not natively available on a vast number of accelerators and architectures.

### 2.3 Existing Solutions and the Low-Rank Paradigm

Given that fine-grained scale granularities fundamentally break the nature of GEMM, a critical question arises: how can we achieve high quantization accuracy while strictly preserving standard matrix multiplication? To address this, the research community has proposed several techniques that attempt to bypass the limitations of per-channel quantization without relying on custom dequantization kernels.

The most popular post-training quantization methods operate directly on the weight distribution itself. GPTQ (Frantar et al., 2023), given a small calibration set, derives the curvature of the output error with respect to the weights and sequentially compensates for the quantization error as each column is quantized. Similarly, AWQ (Lin et al., 2024) observes that only a small fraction of weight channels is disproportionately salient for the output and protects these channels through an activation-aware, lossless rescaling. Both methods preserve standard GEMM compatibility at inference time, yet they fundamentally depend on calibration data.

Another approach is to handle outlier values explicitly. For instance, LLM.int8() (Dettmers et al., 2022) isolates high-magnitude outliers into a separate, high-precision matrix. This effectively splits the computation into multiple parallel GEMM operations and introduces noticable memory overhead and gathering operations. Alternatively, rotation-based methods like QuaRot (Ashkboos et al., 2024) multiply weights and activations by orthogonal matrices to evenly distribute outlier magnitudes across all channels. Although this mathematically eliminates outliers, it introduces an additional $O(N \log N)$ computational overhead and requires continuous feature transformations between network layers.

Other recent approaches focus on structural model modifications. SplitQuantV2 (Song & Lin, 2025) enhances basic linear quantization by using clustering to split a single linear layer into three functionally equivalent, quantization-friendly layers. While this approach is hardware-agnostic and avoids custom kernels, it explicitly forces the hardware to execute multiple sequential matrix multiplications per layer and element-wise additions, which inherently increases the overall inference latency and expands the memory footprint of the quantized model.

Table 1 summarizes the practical deployment constraints of the discussed approaches.

Table 1: Practical deployment constraints of hardware-friendly quantization methods.

| **Method** | **Calibration Data?** | **Custom Kernels?** | **Runtime Overhead?** |
|---|---|---|---|
| LLM.int8() | Yes | Yes | Yes |
| AWQ | Yes | No | No |
| GPTQ | Yes | No | No |
| QuaRot | No | No | Yes |
| SplitQuant | No | No | Yes |

Ultimately, all these methods introduce structural compromises - whether through extra computational steps, multiple matrix multiplications, or complex memory management. Consequently, no existing approach improves reconstruction accuracy in a purely tradeoff-free manner: every gain in fidelity is purchased with a list of additional requirements.

Low-rank approximation offers a different paradigm for model compression and efficiency. It has been foundational for parameter-efficient fine-tuning techniques, most notably LoRA (Hu et al., 2022), which updates models using low-rank weight matrices. More recently, low-rank approximation has been introduced to the quantization pipeline by methods such as LQER (Zhang et al., 2024), which reconstructs the quantization error matrix using high-precision low-rank components executed in parallel with the quantized weights.

However, low-rank approximation has historically been applied strictly to the weight matrices or the error matrices. To the best of our knowledge, low-rank factorization has never been utilized to compress and decouple the quantization scale matrix itself. This unexplored intersection forms the foundation of our proposed method.

# 3 METHOD

To achieve the accuracy of fine-grained quantizations while maintaining the computational efficiency of hardware-friendly quantizations, we propose approximating the scale matrix. Empirical analysis demonstrates that emulating strictly per-element quantization (per-group with $g = 1$) yields the highest theoretical reconstruction accuracy and optimally utilizes the target type dynamic range (detailed in Appendix A).

## 3.1 GEMM AND MATRIX FACTORIZATIONS

Let $\mathbf{S} \in \mathbb{R}^{N \times K}$ be the per-element scale matrix for a weight matrix $\mathbf{W} \in \mathbb{R}^{N \times K}$. We define each element as $\mathbf{S}_{i,k} = |\mathbf{W}_{i,k}|/q_{\max}$. To prevent division by zero, we enforce a strict lower bound: $\mathbf{S}^+ = \max(\mathbf{S}, \epsilon)$, where $\epsilon = 10^{-7}$.

We approximate the matrix $\mathbf{S}^+$ using a rank-1 factorization: $\mathbf{S}^+ \approx \mathbf{u}\mathbf{v}^T$, where $\mathbf{u} \in \mathbb{R}^N$ corresponds to the output-channel axis and $\mathbf{v} \in \mathbb{R}^K$ corresponds to the input-channel axis. Under this approximation, the dequantized weights are represented as $\mathbf{W}_q \approx (\mathbf{u}\mathbf{v}^T) \odot \hat{\mathbf{W}}$, where $\odot$ denotes element-wise multiplication.

Substituting this rank-1 scale approximation into the linear projection equation $\mathbf{Y} = \mathbf{W}_q\mathbf{A}$ yields:

$$\mathbf{Y}_{i,j} \approx \sum_{k=1}^{K} \left(\mathbf{u}_i \mathbf{v}_k \hat{\mathbf{W}}_{i,k}\right) \mathbf{A}_{k,j} = \mathbf{u}_i \sum_{k=1}^{K} \hat{\mathbf{W}}_{i,k} \left(\mathbf{v}_k \mathbf{A}_{k,j}\right). \tag{5}$$

By defining a pre-scaled activation matrix $\tilde{\mathbf{A}}$ such that $\tilde{\mathbf{A}}_{k,j} = \mathbf{v}_k \mathbf{A}_{k,j}$, the entire operation simplifies to:

$$\mathbf{Y} = \text{diag}(\mathbf{u}) \left(\hat{\mathbf{W}}\tilde{\mathbf{A}}\right). \tag{6}$$

This factorization is the mathematical core of the algorithm. The reduction-axis scales $\mathbf{v}$ are folded into the activations before the matrix multiplication, which requires merely a vector scaling of the $K \times M$ activation matrix ($O(KM)$ operations, i.e., $O(K)$ per token). The core operation, $\hat{\mathbf{W}}\tilde{\mathbf{A}}$, is executed as a single standard GEMM. Finally, the output-axis scales $\mathbf{u}$ are applied via a vector scaling of the $N \times M$ output ($O(NM)$ operations, i.e., $O(N)$ per token). The memory overhead for metadata is reduced to just $N + K$ parameters.

## 3.2 UPPER-BOUNDING RANK-1 DECOMPOSITION

The standard approach to low-rank matrix approximation is Truncated Singular Value Decomposition (SVD), which minimizes the Frobenius norm $\|\mathbf{S}^+ - \mathbf{u}\mathbf{v}^T\|_F^2$. While optimal for signal-to-noise separation, Frobenius minimization allows the approximation to underestimate the target matrix ($\mathbf{u}_i\mathbf{v}_k < \mathbf{S}^+_{i,k}$). In the context of quantization, an underestimated scale causes the corresponding weight to exceed the integer dynamic range, resulting in severe weight clipping. Given that LLMs are highly sensitive to the truncation of large-magnitude outliers, traditional SVD degrades model accuracy significantly.

To guarantee that no weight clipping occurs, LRSR imposes a strict element-wise upper-bound constraint:

$$\mathbf{u}_i \mathbf{v}_k \geq \mathbf{S}^+_{i,k} \quad \forall i, k. \tag{7}$$

This formulation ensures that the reconstructed scale always encapsulates the local dynamic range. We construct such a factorization with a closed-form two-step procedure (Algorithm 1).

The decomposition admits a natural interpretation as a principled refinement of per-channel quantization. The first step fixes $\mathbf{u}$ to the exact per-channel upper bounds $\mathbf{c}_i = \max_k \mathbf{S}^+_{i,k}$. The second step computes the smallest reduction-axis factors $\mathbf{v}$ that preserve the elementwise upper-bound constraint. Since $\mathbf{S}^+_{i,k} \leq \mathbf{c}_i$, we have $\mathbf{v}_k \leq 1$, and therefore

$$\mathbf{S}^+_{i,k} \leq \mathbf{u}_i \mathbf{v}_k \leq \mathbf{c}_i = \max_k \mathbf{S}^+_{i,k}. \tag{8}$$

**Algorithm 1** Upper-Bounding Rank-1 Decomposition

**Require:** Per-element scale matrix $\mathbf{S}^+ \in \mathbb{R}^{N \times K}$
**Ensure:** Rank-1 factors $\mathbf{u} \in \mathbb{R}^N, \mathbf{v} \in \mathbb{R}^K$ such that $\mathbf{u}\mathbf{v}^T \geq \mathbf{S}^+$

1: $\mathbf{u}_i \leftarrow \max_k \left(\mathbf{S}^+_{i,k}\right) \quad \forall i$ ▷ Row-wise maxima
2: $\mathbf{v}_k \leftarrow \max_i \left(\frac{\mathbf{S}^+_{i,k}}{\mathbf{u}_i}\right) \quad \forall k$ ▷ Column-wise maxima
3: **return** $\mathbf{u}, \mathbf{v}$

Thus per-channel quantization is the anchor of LRSR, and the vector $\mathbf{v}$ provides an additional degree of freedom along the reduction axis: it tightens the per-channel scale budget at those input channels where this can be done without violating the upper-bound constraint.

**Proposition 1.** Let $\mathbf{c}_i = \max_k \mathbf{S}^+_{i,k}$ and let $(\mathbf{u}, \mathbf{v})$ be produced by Algorithm 1. Then $\mathbf{u} = \mathbf{c}$, $\mathbf{S}^+ \leq \mathbf{u}\mathbf{v}^T \leq \mathbf{c}\mathbf{1}^T$, and $\mathbf{v}$ is the componentwise smallest vector such that $\mathbf{c}_i\mathbf{v}_k \geq \mathbf{S}^+_{i,k}$ for all $i, k$. Consequently, in the standard round-to-nearest no-clipping regime, the elementwise dequantization error bound is no larger than the corresponding per-channel bound.

*Proof.* By construction,

$$\mathbf{v}_k = \max_i \frac{\mathbf{S}^+_{i,k}}{\mathbf{c}_i},$$

so $\mathbf{c}_i\mathbf{v}_k \geq \mathbf{S}^+_{i,k}$ for every $i, k$. Since $\mathbf{S}^+_{i,k} \leq \mathbf{c}_i$, we also have $\mathbf{v}_k \leq 1$, which gives $\mathbf{u}_i\mathbf{v}_k \leq \mathbf{c}_i$. Any feasible vector $\tilde{\mathbf{v}}$ satisfying $\mathbf{c}_i\tilde{\mathbf{v}}_k \geq \mathbf{S}^+_{i,k}$ must obey $\tilde{\mathbf{v}}_k \geq \mathbf{S}^+_{i,k}/\mathbf{c}_i$ for every $i$, hence $\tilde{\mathbf{v}}_k \geq \mathbf{v}_k$. Finally, under round-to-nearest quantization without clipping,

$$|\mathbf{W}_{i,k} - \mathbf{W}^q_{i,k}| \leq \frac{\mathbf{u}_i\mathbf{v}_k}{2} \leq \frac{\mathbf{c}_i}{2}.$$

In the worst case, when no reduction-axis channel can be safely discounted ($\mathbf{v}_k = 1$ for all $k$), the algorithm degenerates to per-channel quantization. In the best case, when the scale matrix $\mathbf{S}^+$ is exactly rank-1, the algorithm recovers full per-element quantization with zero approximation error. Practical weight matrices lie between these extremes, with the approximation error governed by how far the scale matrix is from being rank-1.

As visually demonstrated in Figure 1, this algorithm covers the original scale values with a rank-1 "tent". Unlike per-channel quantization, which projects a coarse global upper bound across the entire axis, LRSR dynamically adapts to the local geometry of the weights, significantly minimizing the error, though not as greatly as per-group quantization.

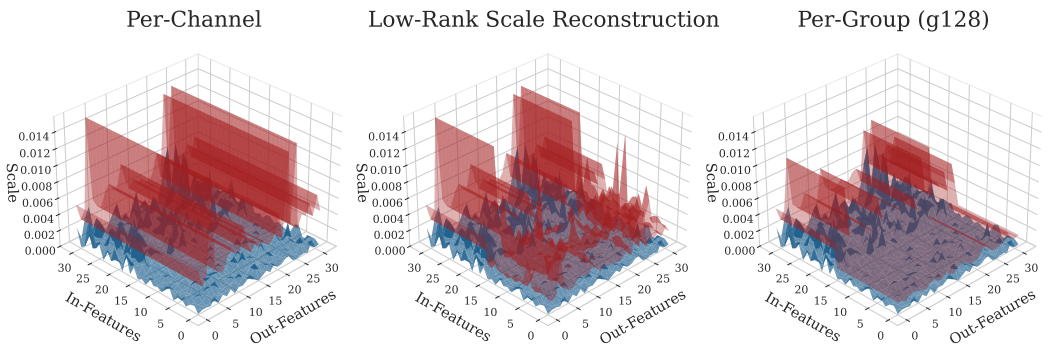



Figure 1: 3D Surface Analysis of Scale Matrices. Visualization of a representative 32x32 chunk. Comparison of the scale matrices of the different granularities. The blue surface represents the original per-element scales, the red surface represents the applied scale for an element of different methods.

The resulting approximation and computation errors, visualized in Figure 2, confirm that LRSR serves as a highly effective middle ground. By factoring scales into both dimensions, LRSR visibly mitigates the severe, structurally aligned errors inherent to per-channel quantization, achieving signal fidelity that approaches explicitly stored per-group scales.

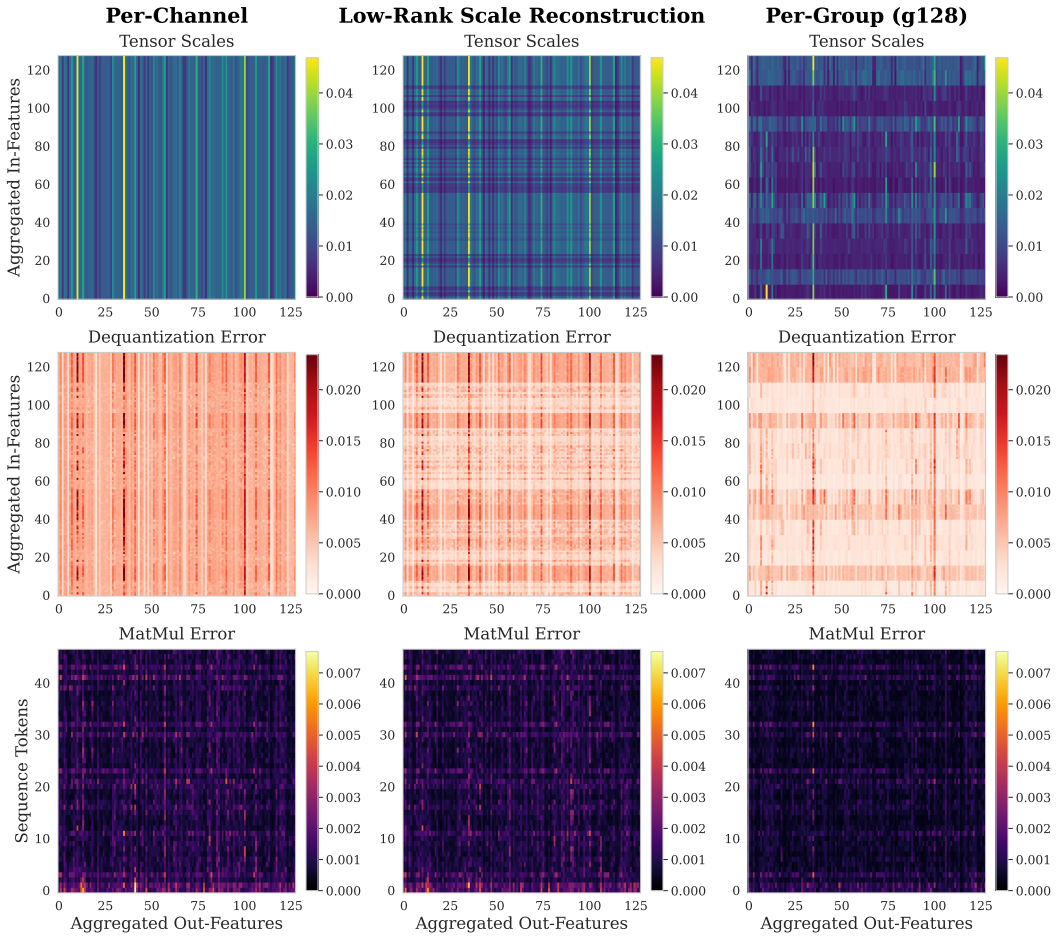



Figure 2: Comprehensive Error Analysis between different granularities. Features on both axes are aggregated into groups via max element. First row: applied scale matrix for each element; second row: weight dequantization error; third row: matrix multiplication error (with prepared activations).

### 3.3 Accuracy Enhancement and Workload Balancing

The mathematical precision of a rank-1 approximation is fundamentally bounded by the linear dependence among the columns (and rows) of the target matrix $\mathbf{S}^+$. This comes from the definition of matrix rank. When a linear layer contains a complex, multi-modal scale distribution, a single global $\mathbf{u}$ and $\mathbf{v}$ pair may introduce excessive upper-bounding error over high-entropy regions. However, this inherent limitation also presents a substantial optimization potential: if we partition the weight matrix into multiple distinct clusters based on their structural similarity, the rank-1 approximation within each cluster becomes substantially more precise.

While dividing a layer into $C$ separate clusters naturally splits the computation into $C$ smaller GEMM operations, the strategy used to define these clusters is critical. A naive chunking approach divides the output features into exactly equal consecutive blocks. While this guarantees perfect hardware load balancing across parallel compute cores, it does so blindly, completely ignoring the structural variance of the scales.

To exploit the linear dependence of the scales without sacrificing the perfect hardware utilization of naive chunking, we formulated two advanced clusterization strategies:

- **1-Dimensional Outlier Separation:** Empirical analysis reveals that the majority of output-feature scale profiles are highly co-directional with the median profile, while a small subset constitutes severe outliers. This method leverages thresholding on angular distances to isolate these outliers, demonstrating that segregating a small fraction of high-entropy output features dramatically tightens the approximation for the rest of the matrix.
- **Constrained K-Means Clustering:** To solve the load-balancing problem combinatorially, we apply a capacity-constrained K-Means algorithm to the scale profiles. This approach optimally permutes and groups structurally similar output features while strictly enforcing that every cluster contains around $N/C$ elements. Consequently, it achieves a great reduction in approximation error compared to naive chunking while guaranteeing a near-identical FLOP distribution per compute core.

Because clusterization inherently shifts the computational profile from a single GEMM to multiple GEMMs, directly comparing it against single-GEMM methods in end-to-end metrics is unfair. Therefore, in the primary experimental section (Section 4), we strictly evaluate the foundational, single-block LRSR. The comprehensive theoretical proofs, algorithms, and extended benchmarks for these clusterization techniques are detailed extensively as an independent study in Appendix B.

# 4 Experiments

To evaluate the practical viability of Low-Rank Scale Reconstruction (LRSR), we prioritize end-to-end generative metrics and direct hardware profiling. Our primary evaluation focuses on the statistical degradation of the model's output distribution and the exact runtime/memory overhead during inference.

The Qwen3 family of models (1.7B and 4B parameters) under INT4 and INT8 weight-only post-training quantization was chosen for benchmarks. To rigorously assess generative degradation, we perform a forward pass (64 continuous chunks of 512 tokens each) over the WikiText-2 test split (`wiki.test.raw`). We evaluate the divergence between the output token distributions of the unquantized BF16 baseline ($P_{\text{base}}$) and the quantized model ($P_{\text{quant}}$) via the Kullback-Leibler Divergence:

$$D_{\text{KL}}(P_{\text{base}} \parallel P_{\text{quant}}) = \frac{1}{M} \sum_{m=1}^{M} \sum_{v \in \mathcal{V}} P_{\text{base}}(v \mid x_{<m}) \log \left( \frac{P_{\text{base}}(v \mid x_{<m})}{P_{\text{quant}}(v \mid x_{<m})} \right), \tag{9}$$

where $\mathcal{V}$ denotes the vocabulary and $M$ is the total evaluated token count. Additionally, we compute the Top-1% token agreement to measure the exact greedy next-token prediction consistency:

$$\text{Top-1} = \frac{1}{M} \sum_{m=1}^{M} \mathbb{I} \left( \arg\max_{v \in \mathcal{V}} P_{\text{quant}}(v \mid x_{<m}) = \arg\max_{v \in \mathcal{V}} P_{\text{base}}(v \mid x_{<m}) \right) \times 100\%, \tag{10}$$

where $\mathbb{I}(\cdot)$ represents the indicator function.

For hardware profiling, we utilize a controlled CUDA-graph replay pipeline. We measure prefill (sequence length 4096) and decode (over 128 tokens) wall-clock latency, as well as peak memory allocation on a single consumer GPU. We benchmark our proposed LRSR algorithm against strict per-channel and per-group ($g = 128$) baselines.

Exhaustive experimental details, extended metrics (e.g., Perplexity, Probability Difference) and the performance of introduced clusterizations are documented in Appendix C.

## 4.1 End-to-End Generative Quality

Figures 3 and 4 illustrate the end-to-end generative degradation of the Qwen3-4B model at INT4 precision.

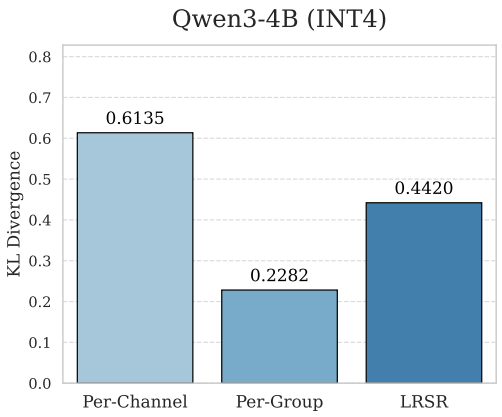



Figure 3: Mean KL-Divergence against BF16. Lower is better.

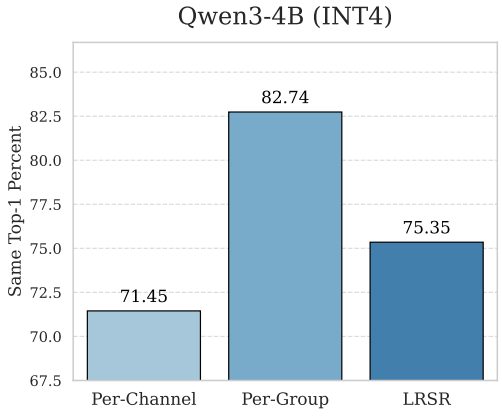



Figure 4: Top-1% Token Agreement with BF16. Higher is better.

As expected, per-channel quantization suffers significant degradation at INT4, exhibiting the highest KL-Divergence and lowest token agreement. Per-group quantization provides a strong accuracy ceiling. The proposed LRSR recovers a substantial portion of the lost accuracy, serving as a powerful middle ground.

### 4.2 Computational and Memory Efficiency

The results of the inference latency and memory footprint benchmark for Qwen3-4B at INT4 are summarized in Table 2.

Table 2: Hardware Profiling of Qwen3-4B at INT4.

| **Method** | **Prefill (ms)** | **Decode (ms)** | **Memory (MB)** | **Overhead (MB)** |
|---|---|---|---|---|
| Per-Channel | 1648.9 | 4257.9 | 2563.5 | baseline |
| Per-Group ($g = 128$) | 1657.5 | 4319.4 | 2615.6 | +52.1 |
| LRSR | 1657.1 | 4321.7 | 2565.3 | +1.8 |

The latency of LRSR is comparable to both baselines. The additional computation required by LRSR - a per-token $O(K)$ vector scaling folded into the activations - is negligible relative to the primary $O(N \times K)$ GEMM operation and does not change the observed latency.

The most critical result lies in the memory footprint. Per-group quantization ($g = 128$) requires an additional 52.1 MB of explicit scale metadata to maintain its accuracy. In contrast, LRSR requires only the $\mathbf{u}$ and $\mathbf{v}$ vectors, accumulating a mere 1.8 MB of metadata overhead.

### 4.3 The Pareto Frontier of Quantization

By unifying the generative quality metrics and the hardware profiling data, we construct the Pareto frontier of post-training quantization expediency, illustrated in Figure 5.

The visualization clearly demonstrates the positioning of the methods. Per-channel is highly efficient but inaccurate; per-group is highly precise but requires more memory. LRSR aggressively shifts the curve toward the optimal lower-left quadrant. While LRSR does not perfectly match the absolute accuracy of explicitly stored per-group scales, it delivers a substantially higher accuracy return per megabyte of metadata.

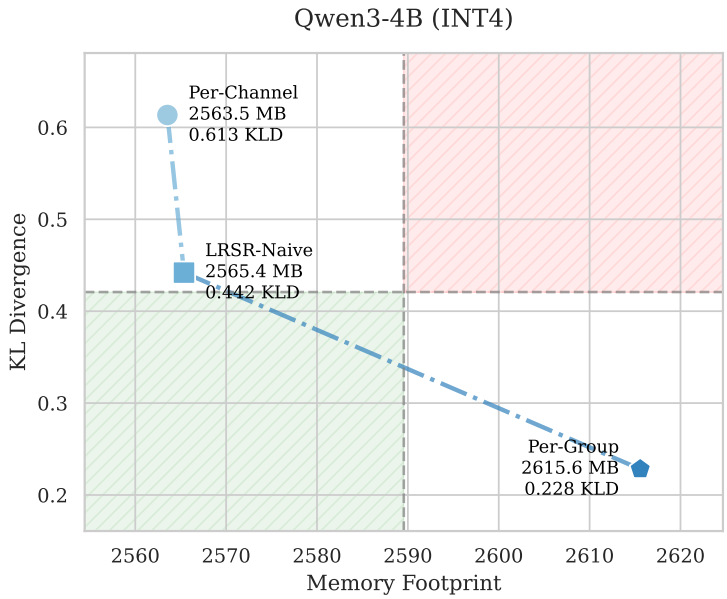



Figure 5: KL-Divergence vs Memory Footprint. The lower-left quadrant represents the best methods, the upper-right the worst.

## 5 Conclusion

In this work, we introduced Low-Rank Scale Reconstruction (LRSR), a novel post-training quantization designed to bridge the gap between hardware-friendly execution and fine-grained quantization accuracy. By approximating the per-element scale matrix through an upper-bounding rank-1 decomposition, LRSR decouples the localized scaling factors into a reduction axis component, which is folded into the activations, and a non-reduction axis component, which is applied to the output. This mechanism enables the execution of fine-grained quantization natively, relying exclusively on standard GEMM operations.

The advantages of this method are numerous. Primarily, LRSR imposes near-zero metadata overhead and requires no calibration data or forward passes, making it exceptionally lightweight. Because it relies entirely on general matrix multiplication, it is natively compatible with the majority of accelerators and architectures. Furthermore, the method is compatible with existing preconditioning techniques - such as orthogonal rotations - allowing for direct composability.

Despite establishing a new Pareto frontier for GEMM-compatible quantization, LRSR possesses certain limitations. First, while the accuracy improvement over per-channel quantization is evident, it remains insufficient for extreme low-bit quantization regimes (e.g., INT2), where explicit outlier handling is typically unavoidable. Second, at higher precisions such as INT8, baseline per-channel quantization is already near-lossless, rendering the relative benefits of LRSR mathematically marginal. Third, injecting the factorization vector into the activation matrix inherently alters the existing activation distribution, potentially degrading its quantization fidelity. We empirically investigate this phenomenon in Appendix D.

Nevertheless, within the strict constraints of calibration-free, custom-kernel-free, and hardware-agnostic environments, LRSR represents a new state-of-the-art in GEMM-compatible model compression. Moreover, the low-rank approximation of the scale matrices possesses intrinsic mathematical properties, which open a new direction for further quantization and optimization research.

### AI USE STATEMENT

In this work, generative AI tools were used for the following tasks with required disclosure: propose or refine hypotheses - the initial research direction surfaced during a brainstorming session with a generative AI tool while solving an applied problem; assist in the writing of proofs - specifically, in formulating the statement of Proposition 1; implement methods - the implementation of the proposed quantization procedure and the benchmark pipeline was partially generated with AI assistance; and assist with translation - the manuscript was drafted in a native language and translated into English with AI assistance, together with adaptation to scientific writing style. Generative AI tools were not used for the generation of synthetic datasets, the development of theoretical models or conceptual frameworks, the formulation of mathematical claims, providing critical ingredients for proving mathematical claims, the design or feedback on the research methodology or experiments, the cleaning or reformatting of datasets, the support of qualitative or thematic data analysis, or the interpretation of results.

Additionally, generative AI tools were used for the following tasks with recommended disclosure: creating and editing software code; brainstorming; sourcing and searching for information; summarizing and analyzing existing literature; identifying relevant literature; and editing the research paper to improve readability and the consistency of its logical structure. We have reviewed all AI-assisted work: every AI-assisted piece of code was manually reviewed and verified by running the full benchmark pipeline and comparing the results with theoretical values, each section of the paper was carefully reviewed to maintain the idea of original draft, and all cited references were checked against the original sources. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI tools.

### ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. The study involves no human subjects, no animal experimentation, and no personally identifiable information, and no new datasets were collected, created, or released. All models and datasets were used in accordance with their respective licenses and usage terms. The proposed method is a post-training quantization; we are not aware of any safety, privacy, fairness, or dual-use concerns arising from its use or deployment. We declare no conflicts of interest or external sponsorship.

### REPRODUCIBILITY STATEMENT

We have made an effort to ensure that the results in this paper are reproducible. The proposed method is fully specified in Section 3. All experimental details - software versions, utilized hardware, the definitions of all reported metrics, and the data and profiling protocol - are described in Appendix C. All models and benchmark datasets used in the experiments are publicly available. The complete source code implementing the method and the benchmark pipeline is provided as an anonymous supplementary material.

## 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Paul S. Bradley, Kristin P. Bennett, and Ayhan Demiriz. Constrained k-means clustering. *Microsoft Research Technical Report MSR-TR-2000-65*, 2000.

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

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Ji Lin, 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. In *Proceedings of Machine Learning and Systems (MLSys)*, 2024.

Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

Jaewoo Song and Fangzhen Lin. Splitquantv2: Enhancing low-bit quantization of llms without gpus. *arXiv preprint arXiv:2503.07657*, 2025.

Cheng Zhang, Jianyi Cheng, George A. Constantinides, and Yiren Zhao. Lqer: Low-rank quantization error reconstruction for llms. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

## A GROUP-SIZE EMULATION AND DYNAMIC RANGE UTILIZATION

The fundamental trade-off in post-training quantization lies between the granularity of the scale factors and the memory overhead required to store them. As the group size $g$ decreases, the localized scales capture the weight distribution more accurately, reducing the quantization error. However, this proportionally increases the metadata footprint. Historically, the research community has pursued smaller group sizes to preserve accuracy, leading to vector quantization paradigms that rely on extremely fine-grained blocks (e.g., $g = 8$) utilizing learned codebooks to implicitly compress the massive scale matrices.

Extrapolating this logic, the theoretical ceiling for quantization accuracy is per-element quantization, where $g = 1$. In a traditional quantization, implementing $g = 1$ is fundamentally impractical. If every individual weight $\mathbf{W}_{i,k}$ has its own explicit scale $\mathbf{S}_{i,k} = |\mathbf{W}_{i,k}|/q_{\max}$, the metadata matrix $\mathbf{S}$ becomes exactly the same size as the original weight matrix $\mathbf{W}$. Furthermore, the actual quantized weights $\hat{\mathbf{W}}$ degenerate entirely into a 1-bit sign matrix, as every element evaluates to exactly $\pm q_{\max}$. This negates the purpose of utilizing a multi-bit integer dynamic range.

However, within the Low-Rank Scale Reconstruction, this paradigm changes. Because the quantization method does not store the full matrix $\mathbf{S}$ explicitly, but rather its rank-1 approximation $\mathbf{S}^+ \approx \mathbf{u}\mathbf{v}^T$, emulating $g = 1$ becomes a viable objective. Consequently, the $\mathbf{W}_{i,k}/(\mathbf{u}_i\mathbf{v}_k)$ is no longer locked to the extreme values of $\pm q_{\max}$. Instead, the actual quantized weights adaptively disperse across the available integer grid to compensate for the rank-1 approximation error.

To empirically investigate this behavior and evaluate the overall fidelity of the reconstructed weights, we extracted and aggregated the `o_proj` linear layers across diverse model architectures (OLMoE-7B-A1B, Gemma4-5B-E2B, SmolLM3-3B, MiniCPM5-1B, and Qwen3-0.6B).

First, to determine the optimal group size for LRSR, we evaluated the Normalized Mean Squared Error (NMSE) of the dequantized weights against BF16 using rank-1 scale matrix targeting various group sizes.

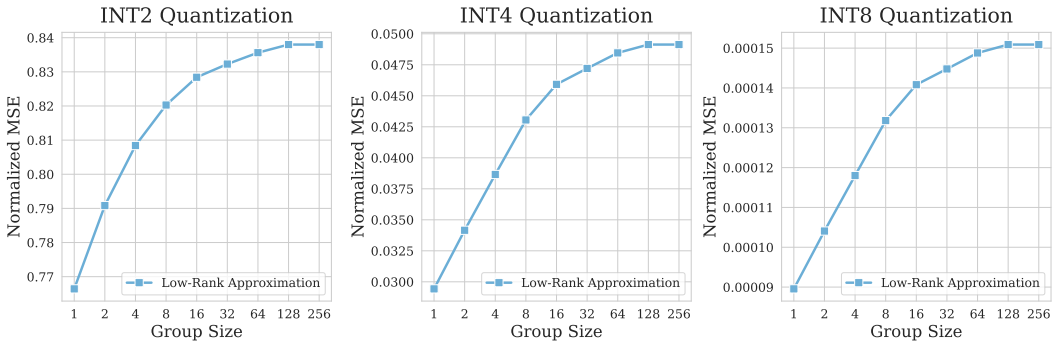



Figure 6: Normalized MSE of the weight dequantization using rank-1 scale approximations when targeting different group sizes.

As shown in Figure 6, minimizing the group size strictly decreases the approximation error across all evaluated bit-widths. The rank-1 decomposition is mathematically most effective when capturing the unaggregated structure of the weight matrix. Therefore, configuring LRSR to target per-element scales ($g = 1$) provides the optimal baseline for the lowest reconstruction error.

Next, we analyze the distribution of the quantized weights within the available integer bins. We evaluated the probability mass function of the quantized weights at INT4 ($[-7, 7]$).

Figure 7 demonstrates that per-channel quantization severely underutilizes the dynamic range, with a substantial portion of weights collapsing to zero due to massive scale inflation caused by isolated outliers. Per-group quantization ($g = 128$) mitigates this issue, yielding a more uniform distribution. The proposed Low-Rank Approximation effectively bridges this gap: by adapting to the approximation error of the $g = 1$ emulation, the weights naturally distribute across the intermediate bins, closely resembling the distribution of the per-group method.

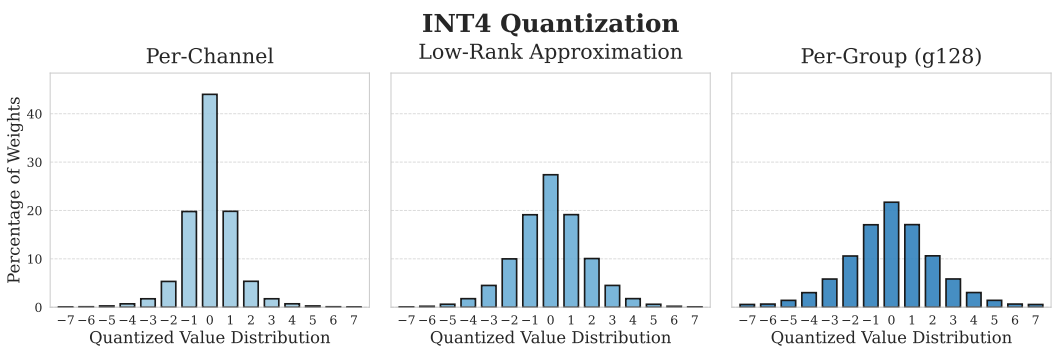



Figure 7: Probability mass function of quantized weights at INT4 for different quantization granularities.

Finally, to benchmark the overall fidelity of the reconstructed weights, we compared LRSR (emulating $g = 1$) against standard per-channel and per-group quantizations across various group sizes.

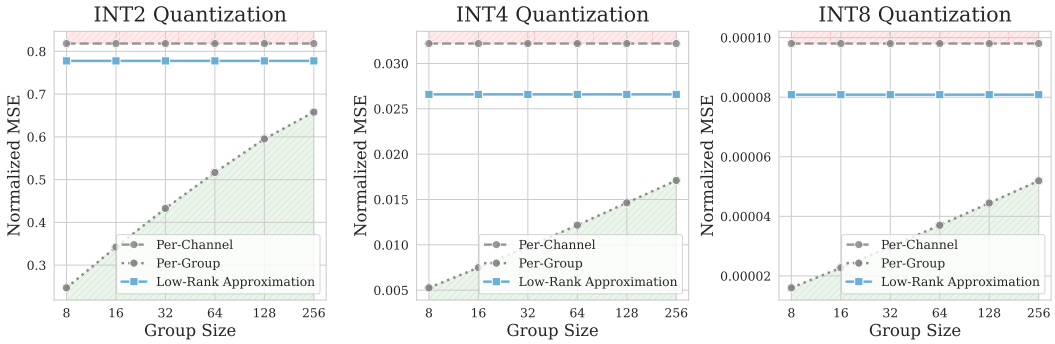



Figure 8: Comparison of Normalized MSE for weight dequantization across different granularities.

As observed in Figure 8, the NMSE of LRSR remains constant and located strictly between the per-channel and per-group baselines. While it does not reach the accuracy of per-group quantization, it consistently outperforms per-channel quantization. As the group size of the per-group method increases, the constant error of LRSR closes an increasingly larger percentage of the gap between the two standard granularities, closing up to 40% of the error gap depending on the bit-width and the chosen baseline group size.

# B Clusterization Analysis and Hardware Load Balancing

## B.1 Theoretical Foundations of Rank-1 Approximability

The core premise of Low-Rank Scale Reconstruction relies on factoring the scale matrix $\mathbf{S}^+$ into two vectors, $\mathbf{u}$ and $\mathbf{v}$. Mathematically, a matrix is perfectly approximable by a rank-1 decomposition (yielding zero approximation error) if and only if all of its rows and columns are strictly linearly dependent. In the context of quantization scales, the closer the output-channel scale profiles of $\mathbf{S}^+$ are to being collinear, the higher the accuracy of the resulting approximation.

To evaluate the degree of linear dependence among the scale vectors, various metrics can be employed, such as the Pearson correlation coefficient or standard cosine similarity. In this study, we utilize the subspace angular distance, which naturally bounds the linear independence regardless of vector magnitudes. For two vectors $\mathbf{x}$ and $\mathbf{y}$, the angular distance $d(\mathbf{x}, \mathbf{y})$ is defined as:

$$d(\mathbf{x}, \mathbf{y}) = 1 - \cos^2(\theta) = 1 - \frac{(\mathbf{x}^T \mathbf{y})^2}{\|\mathbf{x}\|_2^2 \|\mathbf{y}\|_2^2}. \tag{11}$$

A distance of 0 indicates perfect linear dependence (collinearity), while a distance of 1 indicates strict orthogonality.

To understand the structural variance within a linear layer, we compute the median output-channel profile of the scale matrix $\mathbf{S}^+$ and calculate the pairwise angular distance between this median profile and every individual output-channel profile in the matrix. The resulting distribution reveals the underlying entropy of the scale factors.

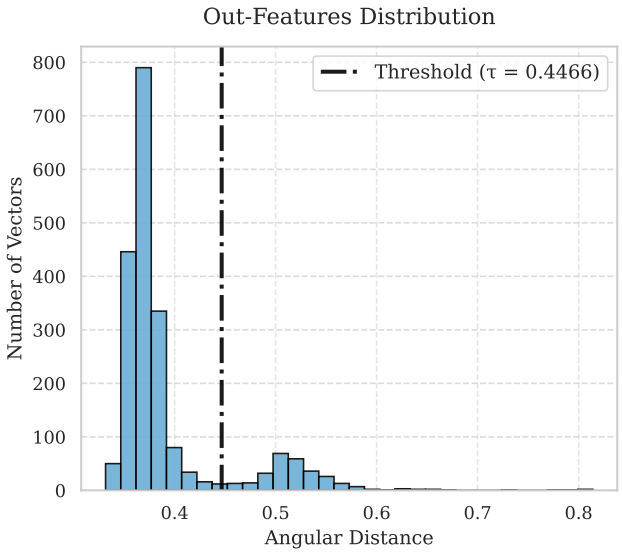



Figure 9: Distribution of angular distances between individual output-channel scale profiles and the median output-channel profile in the `o_proj` layer. The dashed line represents the optimal separation threshold ($\tau$).

As observed in Figure 9, the angular distances do not follow a simple unimodal distribution. Instead, the histogram exhibits a distinct bimodal pattern. The vast majority of the output-channel profiles are densely concentrated at around 0.37, indicating that they share a highly similar structure. However, a secondary peak emerges at higher distances, representing a specific subset of output channels that act as severe structural outliers.

However, not every layer in the network exhibits a bimodal distribution. Experimental observations show that the `gate_proj` layers display a single bell-shaped peak with a heavy right tail rather than two distinct modes. Nevertheless, a significant fraction of the layers in typical architectures does possess the mentioned structure, which is precisely the property we exploit.

This bimodal distribution bears a striking mathematical resemblance to the classic foreground-background segmentation problem in computer vision. It suggests that by isolating these structural outliers into a separate cluster, the rank-1 approximability of the main matrix body could be drastically improved.

To perform this binary classification, we apply Otsu's Method (Otsu, 1979), a variance-minimization algorithm. Otsu's method exhaustively searches for a threshold $\tau$ that minimizes the intra-class variance (or equivalently, maximizes the inter-class variance) between the two resulting groups. It is important to note that while Otsu's method provides a robust, parameter-free mechanism to separate the bimodal peaks, we do not claim it is the definitively optimal thresholding algorithm for quantization scales; rather, it serves as a reliable proof-of-concept to validate our structural separation hypothesis.

### B.2 1-Dimensional Outlier Separation for Accuracy Enhancement

Using the threshold $\tau$ derived from Otsu's method, we perform a 1-Dimensional Outlier Separation (1DOS) on the scale matrix. Specifically, we bifurcate the matrix along the output features axis

into two distinct sub-matrices: a "Major Cluster" containing the highly collinear output-channel profiles ($d < \tau$) and a "Minor Cluster" containing the structural outliers ($d \geq \tau$). We then compute independent rank-1 approximations for each cluster.

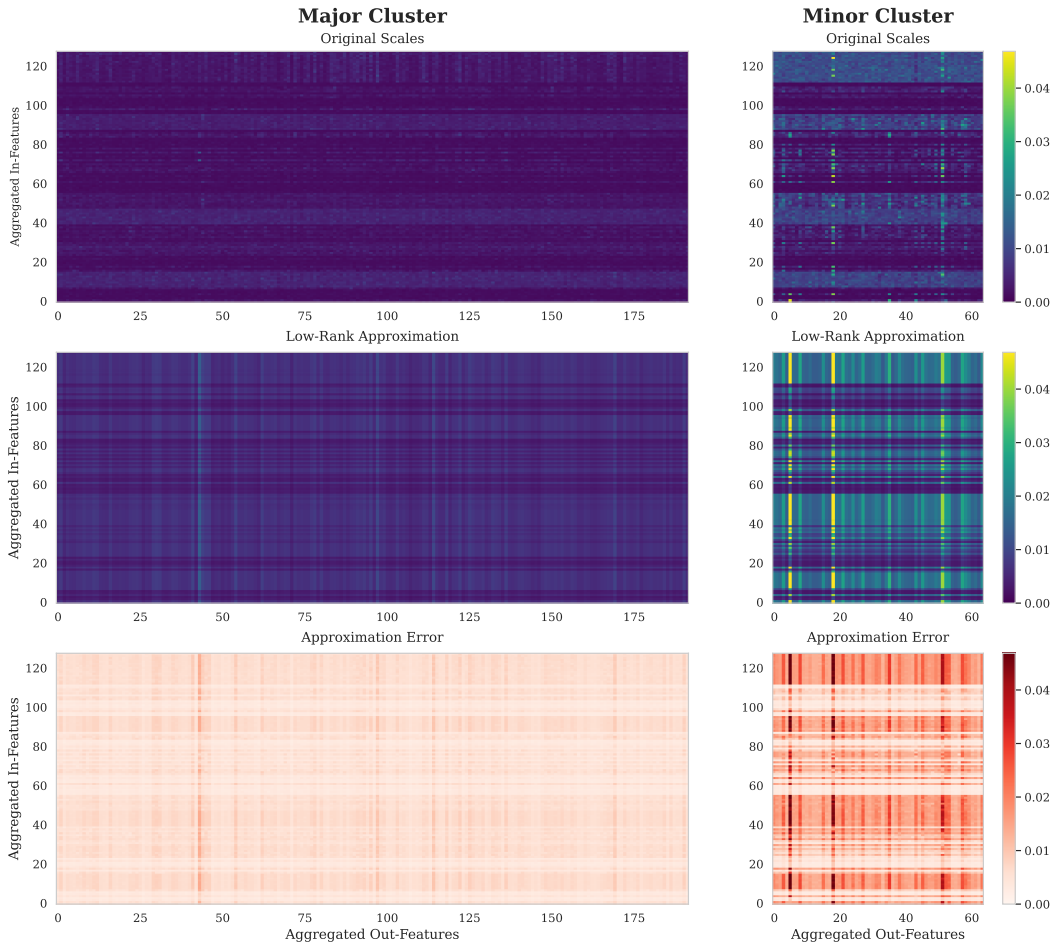



Figure 10: Visualization of 1-Dimensional Outlier Separation. The original scale matrix is divided into a Major Cluster (left) and a Minor Cluster (right).

Figure 10 visualizes the impact of this separation. By isolating the high-entropy output channels, the rank-1 factorization in the Major Cluster is no longer forced to upper-bound extreme variations. Consequently, the approximation error for the vast majority of the weight matrix is substantially minimized.

Theoretically, this approach can be extended to a 2-Dimensional Outlier Separation (2DOS) by simultaneously applying the thresholding technique to the input-channel profiles as well, resulting in four distinct quadrants. To evaluate the efficiency of these structural divisions, we compared the comprehensive dequantization and matrix multiplication errors across three paradigms: the naive single-block method, 1DOS, and 2DOS.

As illustrated in Figure 11, transitioning from a Naive approximation to 1DOS yields a substantial, visually apparent reduction in dequantization error. In contrast, advancing from 1DOS to 2DOS provides only marginal improvements that do not justify the added complexity of managing four distinct computational blocks. Therefore, 1-Dimensional separation is both highly effective and sufficient for maximizing the fidelity of the low-rank approximation.

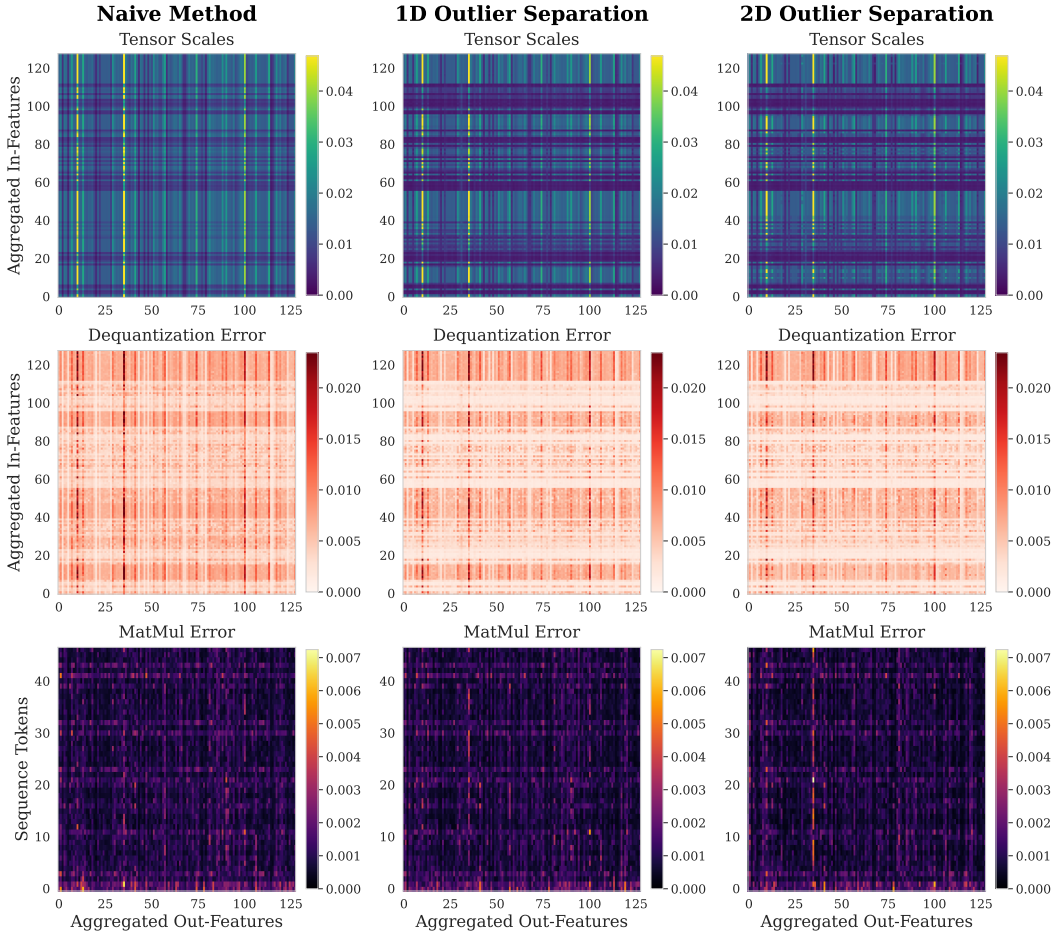



Figure 11: Comprehensive Error Analysis of Outlier Separation methods. Features are aggregated via max element.

A critical advantage of 1DOS is its operational simplicity. Unlike traditional outlier-isolation techniques that rely on sparse matrix multiplications or dequantizations, 1DOS requires no specialized accelerator instructions. The weight matrix is simply sliced into two smaller, dense matrices. Both the major and minor clusters are processed identically using standard GEMM operations, ensuring full compatibility with existing hardware accelerators.

However, despite its mathematical elegance and high reconstruction accuracy, 1DOS introduces a severe limitation in the context of parallel computing. The structural outliers isolated in the Minor Cluster typically account for a very small fraction of the total output channels. Consequently, the two resulting sub-matrices are highly disproportionate in size. When deploying the model on a system with multiple identical compute units, distributing these disproportionate matrices leads to critical workload imbalance. The compute core assigned to the Minor Cluster will finish its GEMM operation almost instantly and remain idle, while the core assigned to the massive Major Cluster becomes the computational bottleneck. This requirement naturally motivates a shift from variance-based thresholding to combinatorial clustering designed for optimal hardware load balancing.

### B.3 Constrained K-Means for Optimal Load Balancing

To partition the scale matrix into $C$ distinct clusters while minimizing intra-cluster variance, standard K-Means clustering is a natural choice. However, standard K-Means purely optimizes for geometric similarity without any constraints on the number of elements assigned to each centroid. As a result,

it frequently produces highly skewed cluster sizes, reintroducing the exact hardware load-balancing bottleneck observed in 1DOS.

To resolve this, we adopt Constrained K-Means (Bradley et al., 2000). This algorithm explicitly enforces a minimum capacity for each cluster during the assignment step. By setting the capacity strictly to $N/C$ (where $N$ is the total number of output features), we guarantee that every resulting sub-matrix will have the exact same dimensions. While the assignment step of Constrained K-Means was traditionally formulated as a computationally expensive Linear Programming problem, modern implementations map this to a Minimum Cost Flow problem on a bipartite graph, making it highly efficient even for large LLM layers.

To validate this approach, we divided the `o_proj` scale matrix into three hardware clusters ($C = 3$) and evaluated the weight dequantization NMSE and the resulting cluster sizes across three strategies: Naive Chunking, Standard K-Means, and Constrained K-Means.

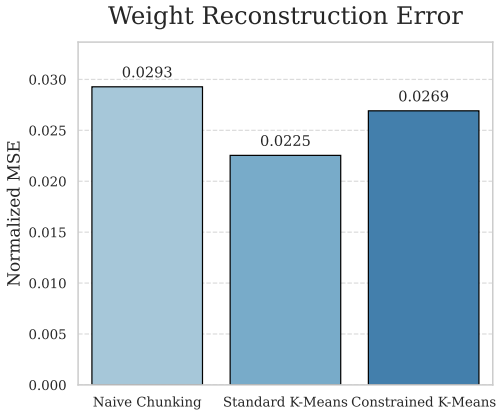



Figure 12: Weight Dequantization NMSE across different clustering strategies ($C = 3$).

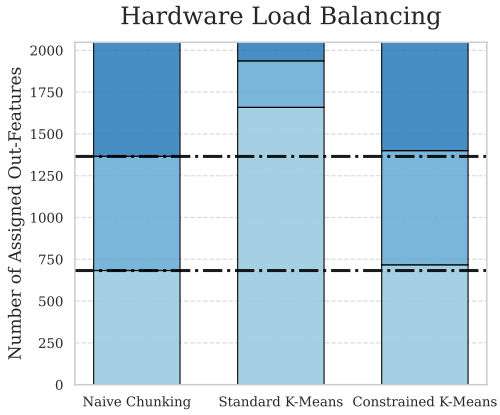



Figure 13: Hardware Load Balancing. Dashed lines indicate the target equal division.

As illustrated in Figures 12 and 13, standard K-Means achieves the lowest reconstruction error but creates severely unbalanced clusters, with one compute core receiving a disproportionately massive workload. Conversely, Naive Chunking provides perfect hardware balancing but suffers from high approximation error due to its blindness to the underlying scale structures.

Constrained K-Means provides the optimal intersection. By combinatorially grouping structurally similar output channels, it noticeably reduces the error compared to the naive approach. Simultaneously, it guarantees an almost equal FLOP budget per cluster, so that identically provisioned compute units receive identically sized sub-problems during parallel GEMM execution.

This demonstrates that the rank-1 approximation of LRSR can be systematically optimized for multi-core hardware topologies without sacrificing compute efficiency. The full end-to-end generative evaluations of these clusterization techniques on the target LLMs are detailed in Appendix C.

## C Comprehensive Benchmarks and Extended Metrics

To ensure reproducibility and provide a rigorous evaluation of the proposed Low-Rank Scale Reconstruction (LRSR) quantization, we conduct an extensive suite of benchmarks. The evaluation is divided into three distinct phases: a layer-wise error analysis, a generative distribution assessment, and exact hardware profiling.

**Environment Setup:** All experiments, data generation, and profiling were conducted using the PyTorch framework (2.13.0+cu130) combined with HuggingFace Transformers (5.15.0). The hardware profiling phase was executed and measured isolated operations on a single consumer-grade GPU (NVIDIA RTX 5070) using CUDA (13.0).

### C.1 Layer-Wise Error Analysis

Before evaluating the end-to-end impact on the LLM, it is crucial to understand how LRSR and its clusterization variants perform at the level of individual linear projections. As established in recent quantization literature, matrix-level Mean Squared Error (MSE) does not perfectly correlate with true generative degradation. Therefore, we utilize this layer-wise benchmark primarily to validate the structural hypotheses of our approximation methods.

**Evaluation Setup:** To construct a highly representative dataset, we extracted the full precision (BF16) weights $\mathbf{W}$ and activation profiles $\mathbf{A}$ from five diverse LLM architectures: Qwen3-0.6B, MiniCPM5-1B, SmolLM3-3B, Gemma4-5B-E2B, and the OLMoE-7B-A1B. For each model, we hooked five critical linear layers (`q_proj`, `k_proj`, `o_proj`, `gate_proj`, `down_proj`) and captured their activations across five fundamentally different prompt contexts (Wikipedia, Python code, mathematics, simple instructions, and poetry).

For every extracted layer, we computed the target matrix multiplication $\mathbf{Y} = \mathbf{W}_q\mathbf{A}$ using INT4 precision across five paradigms: per-channel, per-group ($g = 128$), Naive LRSR, LRSR-1DOS, and LRSR-KMeans ($C = 3$). To evaluate the relative expediency of our methods, we measured the Gap Closed metric, defined as the percentage of the Normalized MSE (NMSE) gap between the hardware-friendly per-channel baseline ($0\%$) and the per-group baseline ($100\%$) that is recovered by LRSR.

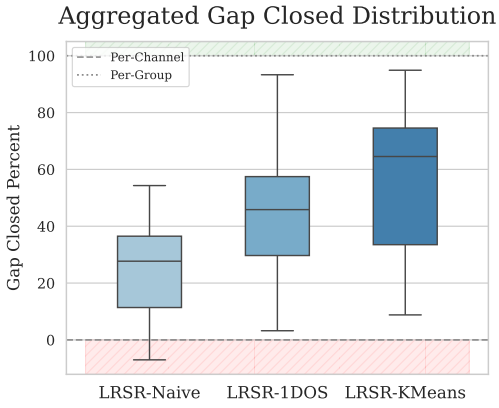



Figure 14: Aggregated distribution of the Gap Closed metric across all tested models, layers, and prompts at INT4.

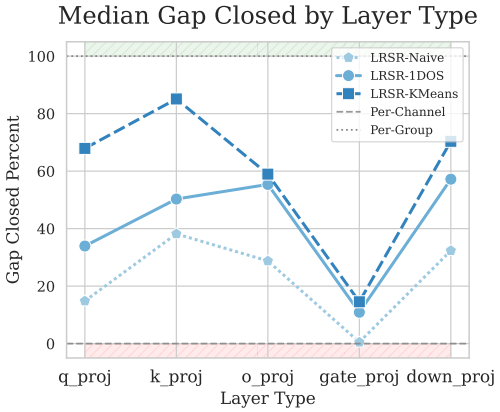



Figure 15: Median Gap Closed stratified by the specific linear layer type, highlighting structural differences.

As shown in Figure 14, naive LRSR serves as a solid baseline, closing a median of approximately 28% of the error gap. By merely isolating high-entropy output channels, LRSR-1DOS pushes the median gap closed to nearly 45%. Finally, constrained K-Means clustering (LRSR-KMeans) demonstrates superior structural adaptation, closing roughly 62% of the gap, with the upper quartile of layers nearing 90% recovery.

Figure 15 provides deeper insight into how different neural network structures react to low-rank scale approximation. The `k_proj` layers exhibit exceptional approximability, with LRSR-KMeans recovering over 80% of the error gap. In contrast, the `gate_proj` layers, which traditionally act as non-linear gating mechanisms and contain highly erratic, non-collinear scale distributions, present a significant challenge. Naive LRSR and 1DOS struggle to outperform per-channel quantization on these specific layers (closing near $0\%$ of the gap). However, the combinatorial grouping of LRSR-KMeans successfully identifies the hidden collinear structures even within the `gate_proj`, recovering nearly 18% of the gap and demonstrating its robustness across all architectural components.

### C.2 Generative Distribution Assessment

While layer-wise errors confirm the mathematical validity of the rank-1 scale approximation, the exhaustive test of a quantization method is its impact on the end-to-end generative capabilities of the model. To evaluate this, we measure the statistical divergence of the quantized model's output logits against the exact unquantized (BF16) reference model.

**Evaluation Setup:** The benchmark was conducted on the standard `WikiText-2` test set (`wiki.test.raw`). To capture long-context dependencies, the text was tokenized into 64 continuous chunks, each with a sequence length of 512 tokens. For every token position, we extracted the full vocabulary logit distributions from both the BF16 baseline and the quantized variants to compute the comparative metrics.

While Kullback-Leibler Divergence (KLD) and Top-1% Token Agreement are established in Section 4, we introduce two additional granular metrics to evaluate the structural integrity of the generation. First, we measure Perplexity (PPL), which characterizes how well the probability model predicts the ground-truth sequence. It is defined exponentially as:

$$\text{PPL} = \exp\left(-\frac{1}{M}\sum_{i=1}^{M}\log P_\theta(x_i \mid x_{<i})\right), \tag{12}$$

where $x_i$ is the ground-truth token. We report the raw PPL alongside the absolute degradation $\Delta\text{PPL} = \text{PPL}_{\text{quant}} - \text{PPL}_{\text{base}}$. Second, we compute the Probability Difference ($\Delta P$), which tracks the average percentage-point shift in the probability assigned to the ground-truth token. A negative value indicates that the quantization process diminishes the model's confidence in the correct token:

$$\Delta P = \frac{1}{M}\sum_{i=1}^{M}\left(P_{\text{quant}}(x_i \mid x_{<i}) - P_{\text{base}}(x_i \mid x_{<i})\right) \times 100\%. \tag{13}$$

The comprehensive results for the Qwen3-1.7B and Qwen3-4B models at both INT8 and INT4 precisions are detailed in Table 3.

Table 3: Comprehensive End-to-End Generative Metrics on `wiki.test.raw` (64 chunks, 512 tokens).

| Model | Bits | Method | PPL | Δ PPL | $\Delta P$ (%) | KLD | Same Top-1 (%) |
|---|---|---|---|---|---|---|---|
| **Qwen3-1.7B** | 8 | Per-Channel | 16.30 | -0.23 | -0.09 | 0.0057 | 96.94 |
| | | Per-Group ($g = 128$) | 16.55 | +0.01 | -0.01 | 0.0035 | 97.48 |
| | | LRSR-Naive | 16.33 | -0.21 | -0.14 | 0.0054 | 97.22 |
| | | LRSR-1DOS | 16.48 | -0.06 | -0.05 | 0.0037 | 97.37 |
| | | LRSR-KMeans | 16.40 | -0.14 | -0.08 | 0.0040 | 97.41 |
| | 4 | Per-Channel | 27.17 | +10.64 | -8.07 | 0.7374 | 65.52 |
| | | Per-Group ($g = 128$) | 22.25 | +5.72 | -3.54 | 0.3679 | 75.57 |
| | | LRSR-Naive | 20.77 | +4.24 | -6.27 | 0.5087 | 71.91 |
| | | LRSR-1DOS | 18.70 | +2.17 | -5.91 | 0.4182 | 74.41 |
| | | LRSR-KMeans | 19.48 | +2.95 | -4.67 | 0.3919 | 75.38 |
| **Qwen3-4B** | 8 | Per-Channel | 14.09 | +0.09 | 0.00 | 0.0033 | 97.55 |
| | | Per-Group ($g = 128$) | 14.05 | +0.05 | +0.01 | 0.0021 | 98.09 |
| | | LRSR-Naive | 14.04 | +0.04 | -0.02 | 0.0031 | 97.74 |
| | | LRSR-1DOS | 14.04 | +0.04 | -0.01 | 0.0027 | 97.86 |
| | | LRSR-KMeans | 14.01 | +0.01 | -0.02 | 0.0028 | 97.78 |
| | 4 | Per-Channel | 24.97 | +10.97 | -5.76 | 0.6135 | 71.45 |
| | | Per-Group ($g = 128$) | 16.72 | +2.72 | -1.87 | 0.2282 | 82.74 |
| | | LRSR-Naive | 20.25 | +6.25 | -4.14 | 0.4420 | 75.35 |
| | | LRSR-1DOS | 17.91 | +3.91 | -2.92 | 0.3336 | 78.43 |
| | | LRSR-KMeans | 21.32 | +7.32 | -2.87 | 0.3918 | 77.42 |

The metrics from Table 3 provide several critical insights into the behavior of the algorithms:

At INT8 precision, the per-channel quantization is already near-lossless. The Kullback-Leibler Divergence is practically negligible ($< 0.006$), and Top-1 token agreement exceeds 97% across all methods. In this regime, the explicit scale mapping of per-group and the structural rank-1 factorization of LRSR offer marginal improvements.

At INT4, the limitations of standard hardware-friendly quantization become evident. The per-channel method suffers severe degradation, with PPL increasing by over 10 points and ground-truth confidence ($\Delta P$) dropping by up to 8%. The foundational LRSR-Naive algorithm systematically mitigates this collapse, recovering a significant portion of the KLD and Top-1 agreement.

The integration of clusterization techniques shows great results. The 1DOS method effectively limits the upper-bounding approximation error, yielding significant improvements in KLD and pushing Top-1 token agreement remarkably close to per-group. By clustering the weights into 3 equal blocks, LRSR-KMeans systematically outperforms Naive LRSR in distribution matching (KLD) and token confidence ($\Delta P$). On the Qwen3-1.7B model, LRSR-KMeans achieves a Top-1 agreement of 75.38%, practically mirroring the 75.57% achieved by the per-group.

### C.3 Hardware Profiling and Latency

While the generative metrics dictate the theoretical viability of a quantization method, its practical value is determined entirely by its execution latency and memory footprint on real hardware. To rigorously benchmark these characteristics, we utilized the isolated CUDA-graph replay pipeline constructed on the idea of "white-box".

**Evaluation Setup:** We measured the wall-clock latency for both the prefill phase (processing a dense prompt of 4096 tokens) and the decode phase (autoregressively generating 128 tokens, without KV-cache). Additionally, we tracked the absolute peak memory allocated on the GPU during the entire execution. The comprehensive profiling results for both Qwen3-1.7B and Qwen3-4B are presented in Table 4.

Table 4: Comprehensive Hardware Profiling at INT4 precision. Latency is reported as the median wall-clock time over 10 trials.

| Model | Method | Prefill (ms) | Decode (ms) | Memory (MB) | Overhead (MB) |
|---|---|---|---|---|---|
| **Qwen3-1.7B** | Per-Channel | 712.69 | 2433.95 | 1331.4 | baseline |
| | Per-Group ($g = 128$) | 713.03 | 2436.23 | 1351.3 | +19.9 |
| | LRSR-Naive | 691.36 | 2479.02 | 1332.4 | +1.0 |
| | LRSR-1DOS | 739.25 | 2528.51 | 1338.4 | +7.0 |
| | LRSR-KMeans | 857.72 | 2553.67 | 1343.2 | +11.8 |
| **Qwen3-4B** | Per-Channel | 1648.94 | 4257.94 | 2563.5 | baseline |
| | Per-Group ($g = 128$) | 1657.50 | 4319.42 | 2615.6 | +52.1 |
| | LRSR-Naive | 1657.13 | 4321.67 | 2565.3 | +1.8 |
| | LRSR-1DOS | 1838.27 | 4242.16 | 2576.8 | +13.3 |
| | LRSR-KMeans | 1965.47 | 4433.69 | 2586.0 | +22.5 |

The profiling confirms that per-group quantization imposes a significant memory penalty (e.g., +52.1 MB for Qwen3-4B). In contrast, the foundational LRSR-Naive method requires an almost imperceptible +1.8 MB overhead. Splitting the scale matrix into multiple clusters adds only a few megabytes on top of this - at most +22.5 MB for LRSR-KMeans on Qwen3-4B - which remains less than half of the metadata required by the explicit per-group scales.

As shown in the table, the latency of LRSR-Naive is practically identical to the highly optimized per-channel baseline, confirming that folding the reduction-axis scales into the activations incurs negligible overhead. The clustered variants, benchmarked with concurrent dispatch of their per-cluster GEMMs, exhibit the same flat profile: the decode latency stays within 5% of the per-channel baseline on both models. The only systematic cost appears in the prefill phase, where overhead ranges from $+3\%$ to $+21\%$ depending on the model and the clusterization strategy.

### C.4 The Extended Pareto Frontier

To complete the picture, we extend the Pareto frontier established in Section 4 with the clusterized variants of LRSR. Plotting the generative degradation against the exact memory footprint for the Qwen3-1.7B model yields the extended frontier shown in Figure 16.

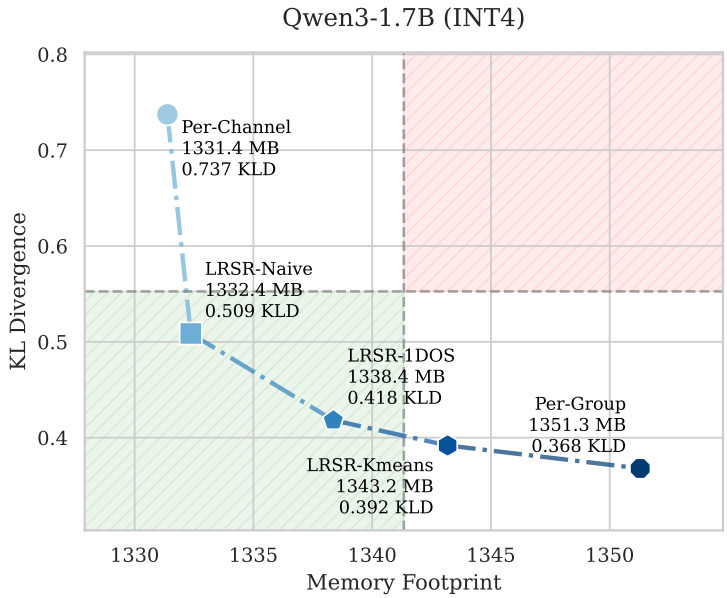



Figure 16: The Extended Pareto Frontier of Quantization Expediency (Qwen3-1.7B at INT4). Lower-left positioning indicates the optimal intersection of high generative fidelity and low memory footprint.

As illustrated in Figure 16, the landscape of hardware-friendly quantization is significantly altered. Per-channel quantization represents the absolute minimum in memory but suffers from severe generative degradation. Per-group quantization achieves high fidelity but demands substantial memory overhead.

The Low-Rank Scale Reconstruction (LRSR) family establishes an entirely new trajectory between these two extremes. The Naive LRSR algorithm immediately yields a massive drop in KLD for a near-zero memory cost. The subsequent application of structural clusterizations smoothly extends this Pareto curve. They allow practitioners to strategically dial in the exact trade-off required for their specific hardware, exchanging just a few megabytes of memory for substantial gains in generative accuracy while utilizing the standard GEMM operation.

Ultimately, LRSR proves that achieving near-per-group quantization fidelity does not require custom kernels, calibration data, or partial dequantization. By exploiting the mathematical properties of rank-1 factorization and combinatorial load balancing, LRSR delivers a highly scalable, hardware-friendly solution optimized for the next generation of rigid hardware accelerators.

## D Analysis of Pre-Scaled Activations

The primary mathematical mechanism of Low-Rank Scale Reconstruction (LRSR) relies on folding the reduction-axis scale vector $\mathbf{v}$ into the input activations prior to the core matrix multiplication: $\tilde{\mathbf{A}}_{k,j} = \mathbf{v}_k \mathbf{A}_{k,j}$. While this element-wise pre-scaling is entirely mathematically lossless in high-bit floating-point precision, it structurally alters the dynamic range of the activation matrix.

This alteration is a non-issue for weight-only quantization paradigms, which are currently the standard for many LLM deployments. However, many specialized hardware accelerators (e.g., Neural Processing Units) natively support only fully quantized integer-to-integer operations (e.g., W8A8 or

W4A4). It is therefore critical to understand how the injection of the vector $\mathbf{v}$ impacts the distribution of the activations and their subsequent quantizability on such constrained devices.

To analyze this, we extracted the reduction-axis vector $\mathbf{v}$ from the upper-bounding decomposition of an `o_proj` layer and evaluated its distribution.

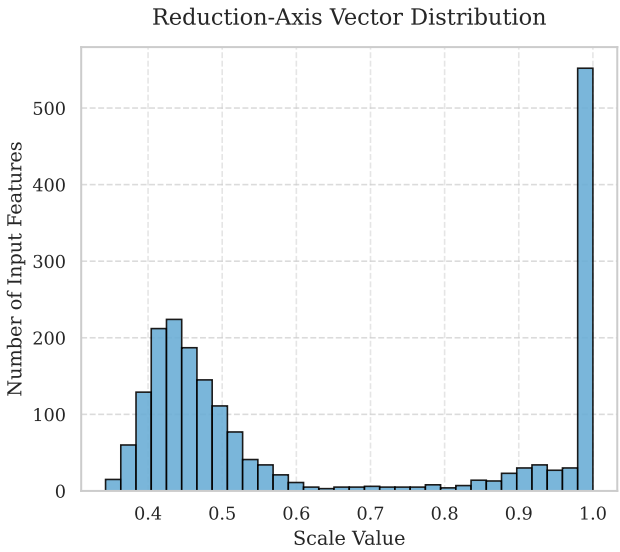



Figure 17: Distribution of the reduction-axis vector ($\mathbf{v}$) values derived from the upper-bounding rank-1 decomposition.

As shown in Figure 17, the values of $\mathbf{v}$ are heavily concentrated exactly at $1.0$, which corresponds to input channels where the per-channel upper-bound could not be safely discounted. The rest of the values form a distinct bell-shaped distribution centered around $0.44$. Because the activations are multiplied by these heterogeneous fractional values, the relative magnitude differences between various input channels are artificially amplified. It should be noted that this distribution is highly layer-dependent; for instance, `down_proj` layers typically exhibit significantly less variance in $\mathbf{v}$, resulting in almost negligible activation perturbation.

To quantify the exact impact of this perturbation on quantization difficulty, we measured the intra-token dynamic range - defined as the ratio of the maximum to the minimum absolute activation value within a single token ($\max_k |\mathbf{A}_{k,j}| / \min_k |\mathbf{A}_{k,j}|$). A larger ratio indicates a wider dynamic range, which generally exacerbates quantization error when mapping to a limited set of integer bins.

Figure 18 plots the Kernel Density Estimation of this metric for both the original and the pre-scaled activations.

The visualization illustrates two distinct statistical shifts caused by the pre-scaling process. First, the primary mode of the distribution shifts to the right, from approximately $10^{5.1}$ to $10^{5.6}$. This indicates that the typical intra-token dynamic range has increased by roughly a factor of $3.2$. This expansion is fundamentally detrimental to activation quantization, as the available integer bins must now stretch over a wider numerical spectrum, leading to a loss of resolution for smaller activation values.

Conversely, the overall width of the distribution noticeably contracts, shrinking from a span of roughly [3.8, 6.4] down to [4.3, 6.3] on the logarithmic scale. This indicates that the tokens have become more homogeneous in terms of their dynamic range. This homogeneity makes the activation statistics more regular, but does not offset the observed shift toward larger dynamic ranges.

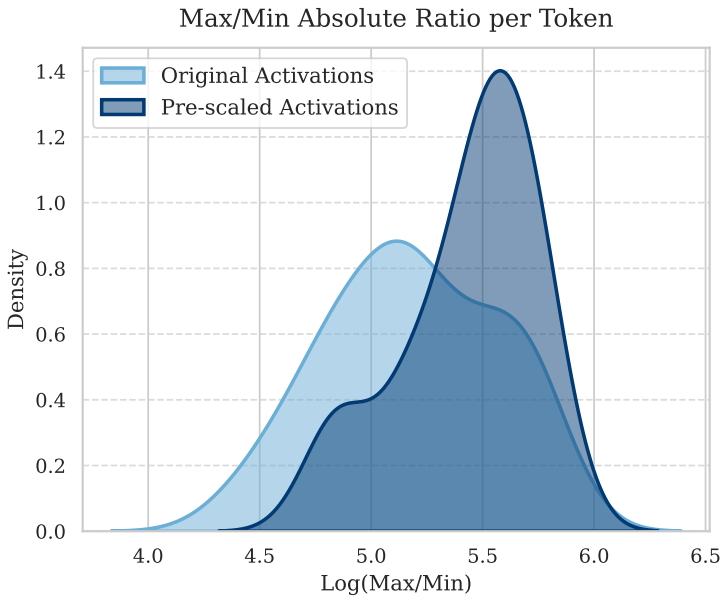



Figure 18: Kernel Density Estimation of the Max/Min absolute ratio per token (in $\log_{10}$ scale), comparing the original activations against the pre-scaled activations.

Ultimately, the impact of LRSR on activations represents a nuanced trade-off. For regimes utilizing high-precision activations (WXA16), the method is mathematically lossless. For moderate activation quantization (WXA8), the increased dynamic range presents a challenge; however, LRSR nevertheless remains viable without additional modifications in many practical settings. However, for extreme symmetric activation quantization (WXA4), the $3.2\times$ expansion in dynamic range may exceed the representational capacity of a 4-bit integer. In such highly constrained scenarios, it is advisable to pair LRSR with other pre-conditioning techniques to smooth the activation outliers prior to applying the reduction-axis vector.