A Direction-Aware Framework for Post-Training Compression

Anonymous Author(s)

Abstract

Post-training mixed-precision compression has largely been driven by magnitude-based objectives, such as reconstruction error, while leaving perturbation direction largely uncontrolled. As a result, compressed models often exhibit scattered and unstable performance shifts: similar levels of compression can lead to markedly different outcomes depending on the induced perturbation. This suggests that compression stability depends not only on perturbation magnitude, but also on perturbation direction relative to the local loss landscape. We propose **DiRa**, a unified, model-agnostic, and training-free framework for **dir**ection-**a**ware post-training compression. DiRa is motivated by a simple view: compression decisions should be evaluated not only by the size of the induced perturbation, but also by its local effect on the loss. To achieve this, DiRa introduces neighborhood-aware local loss surrogates, using a first-order surrogate in N_1 and a second-order surrogate in N_2. Under this unified formulation, DiRa can be instantiated for both tensor decomposition and mixed-precision quantization. Experiments on diverse vision and language models show that DiRa yields more stable compression and competitive or superior accuracy–compression trade-offs compared with existing post-training baselines, while often remaining near-lossless under practical compression settings.

CCS Concepts

• **Computing methodologies** → **Artificial intelligence**.

Keywords

Stable Compression, Mixed-Precision Compression, Direction

ACM Reference Format:

. 2026. A Direction-Aware Framework for Post-Training Compression. In . ACM, New York, NY, USA, 10 pages.

1 Introduction

With the rapid growth in the scale and complexity of deep neural networks, memory and computational costs have become major obstacles to efficient deployment. Model compression has therefore become an essential technique for accelerating inference and reducing deployment cost. Among the many compression paradigms, post-training compression is particularly attractive because it requires little or no retraining, introduces minimal additional overhead, and remains readily compatible with existing training pipelines.

Within this paradigm, mixed-precision compression is especially appealing because it assigns different compression levels to different layers, such as varying quantization bit-widths or decomposition ranks according to layer sensitivity. Compared with uniform compression, this heterogeneous strategy offers greater flexibility in balancing efficiency and accuracy, and often achieves better accuracy–efficiency trade-offs under the same deployment budget.

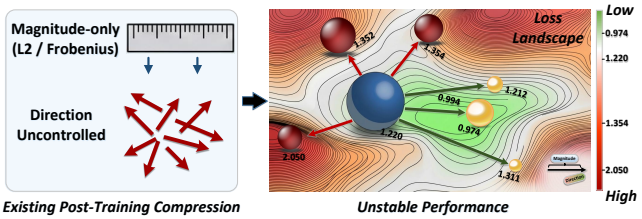

Figure 1: Traditional post-training compression controls only noise magnitude (left), leaving direction uncontrolled, causing instability. DiRa aligns noise with the optimal direction (green arrows), guiding the model (blue sphere) to lower loss (e.g., 0.974) for stable performance (yellow spheres).

As a result, post-training mixed-precision compression has become an increasingly important direction for practical model deployment.

However, existing post-training mixed-precision methods [4, 5, 13, 23, 24, 43] largely focus on minimizing noise magnitude, such as quantization error or reconstruction error, while paying limited attention to noise direction. This limitation is subtle but important. Under a local loss approximation, the effect of a compression perturbation depends not only on its magnitude, but also on its direction in parameter space. Consequently, perturbations with similar norms may induce very different changes in loss, depending on whether they are aligned with favorable or unfavorable directions of the local landscape. In other words, controlling only how much perturbation is introduced is not sufficient; one must also consider how that perturbation interacts with the loss surface.

Recent advances still mainly improve post-training compression through magnitude- or sensitivity-related objectives. For example, SliM-LLM [14] reduces reconstruction error based on salience, ResQ [31] suppresses outliers through low-rank residuals and rotation, and GMPQ-TE [20] uses topological entropy for sensitivity-aware allocation. Despite their differences, these methods do not explicitly control the directional effect of compression noise on local loss. As a result, compression outcomes can remain inconsistent: some perturbations cause noticeable degradation, while others with similar magnitude lead to much smaller loss increase or even mild improvement. As illustrated in Figure 1, this inconsistency suggests that the missing factor in post-training compression is not only perturbation magnitude, but also perturbation direction.

Motivated by this observation, we propose **DiRa**, a unified, model-agnostic, and training-free compression framework for both tensor decomposition and mixed-precision quantization. The central idea of DiRa is to combine neighborhood-aware local loss surrogates with direction-aware perturbation control, so that compression decisions are guided not merely by perturbation size, but by their induced local loss effect. Specifically, within perturbation neighborhoods where local approximation remains reliable, DiRa favors compression decisions whose induced perturbations yield more favorable first-order effects, and when necessary, more favorable joint first- and second-order effects. In this way, DiRa explicitly

Conference'17, Washington, DC, USA

models and mitigates the instability caused by uncontrolled perturbation direction, leading to stable and often near-lossless compression across a wide range of vision and language models, including convolution, vision transformers, and large language models.

Our contributions are summarized as follows:

• We identify perturbation direction as an important but underexplored factor in compression, and show through loss analysis that compression stability depends not only on perturbation magnitude but also on its alignment with the loss landscape.

• We propose DiRa, a unified and training-free direction-aware compression framework that applies neighborhood-aware local loss to both tensor decomposition and mixed-precision quantization.

• We conduct extensive experiments on vision and language models, showing that DiRa delivers more stable compression behavior and competitive or superior accuracy–compression trade-offs compared with existing post-training baselines.

2 Related Works

Post-training compression methods, including tensor decomposition and mixed-precision quantization, are typically guided by magnitude-based surrogate objectives. Representative formulations minimize reconstruction error in parameter space, e.g.,

$$\min \|\mathbf{W} - \hat{\mathbf{W}}\|_F^2 \quad \text{or} \quad \min \|\mathbf{W} - \hat{\mathbf{W}}\|_2^2, \tag{1}$$

where $\hat{\mathbf{W}}$ denotes the compressed approximation under a given compression scheme. While these objectives effectively control perturbation magnitude, they leave the induced perturbation direction largely implicit. As a result, two perturbations with similar magnitude may have very different effects on the loss, leading to inconsistent compression behavior across layers, models, and tasks.

2.1 Tensor Decomposition.

Conventional decomposition methods such as SVD, CP, and Tucker obtain low-rank approximations by minimizing reconstruction error, e.g., $\min \|\mathbf{W} - \mathbf{L}\mathbf{R}^\top\|_F^2$. In post-training settings, such approximations can still incur noticeable accuracy degradation because reconstruction quality alone does not fully characterize the downstream effect of the induced perturbation on the task loss [38–40, 44]. To mitigate this issue, subsequent works have introduced feature-map reconstruction, training-time regularization, or factor reordering strategies, such as DAC [21] and MAESTRO [12]. More recent advances further explore distribution-aware decomposition [17], decomposition for MoE expert weights, and QSVD [35] for efficient QKV compression. However, these methods are still primarily driven by reconstruction fidelity or related surrogates, rather than explicitly modeling the directional effect of the induced perturbation on the loss.

2.2 Mixed-Precision Quantization.

Post-training quantization is also commonly formulated through magnitude- or sensitivity-based objectives, such as minimizing reconstruction error $\|\mathbf{W} - Q(\mathbf{W})\|_2^2$ or using activation-aware scaling and salience estimates to guide per-layer bit-width allocation [4, 8, 23]. Early automated approaches such as HAQ [33] and AutoQ [26] rely on reinforcement learning or sensitivity heuristics; AdaRound [28] and BRECQ [22] further optimize rounding or block-level reconstruction surrogates beyond naive nearest rounding, while more recent techniques continue to improve allocation quality through stronger surrogates: ResQ [31] combines low-rank residuals and rotation to suppress outliers.

Some post-training quantization methods already partially affect perturbation direction, for example through optimized rounding, Hessian-aware scoring, or rotation-based preprocessing. However, they still primarily optimize reconstruction or sensitivity. DiRa differs in explicitly treating perturbation direction within the feasible compression set as a decision variable, under a shared local-loss criterion across quantization and decomposition.

3 Direction-Aware Analytical Framework

3.1 Local Loss Shift Under Compression

Consider an n-layer neural network with parameters $w = (w_1, \ldots, w_n)$ and empirical loss

$$L(w) = \frac{1}{m} \sum_{(x_i, y_i) \in \mathcal{D}} \ell(f(x_i; w), y_i), \tag{2}$$

where $\mathcal{D}$ is the dataset, m is the number of samples, and ℓ denotes the per-sample loss. Let $\hat{w} = w + \delta$ denote the compressed model, where δ is the compression-induced parameter perturbation. The corresponding loss shift is $\Delta L = L(w + \delta) - L(w)$.

Under a local Taylor expansion around w, we have

$$\Delta L \approx \nabla L(w)^\top \delta + \frac{1}{2} \delta^\top H(w) \delta + R_3(\delta), \tag{3}$$

where $H(w)$ is the Hessian of L at w, and $R_3(\delta)$ denotes the higher-order remainder. The first-order term $\nabla L(w)^\top \delta$ captures the directional effect of the compression-induced perturbation, while the second-order term reflects the influence of local curvature. For activation quantization, we use the same principle and treat the induced activation perturbation analogously at corresponding layer.

To favor stable performance after compression, we seek perturbations that induce a non-positive local loss shift, i.e., $\Delta L \lesssim 0$ under the local approximation. The linear term can be made negative by aligning the perturbation direction with the negative gradient ($\nabla L^\top \delta < 0$). However, this Taylor expansion holds only when the perturbations stay within an appropriate range. To formalize this constraint and ensure the differential approximation remains reliable, we define a perturbation neighborhood that bounds the discrepancy between the true loss change and its approximation.

Definition 1 (Perturbation Neighborhood). *For a prescribed radius $\rho > 0$ and approximation tolerance $\eta(\rho) > 0$, we define the perturbation neighborhood as*

$$N_{\rho,\eta} = \left\{ \delta : \|\delta\| \le \rho, \left| \Delta L - \left(\nabla L(w)^\top \delta + \frac{1}{2} \delta^\top H(w) \delta \right) \right| \le \eta(\rho) \right\}. \tag{4}$$

Within $N_{\rho,\eta}$, the first- and second-order terms provide a locally reliable approximation to the true loss change. The radius ρ therefore characterizes the perturbation scale within which direction-aware compression decisions can be evaluated using local loss surrogates. In practice, for each layer we estimate ρ by a discrete perturbation sweep over multiple perturbation directions and calibration samples, and instantiate the tolerance in Eq. (4) by a fixed threshold $\eta(\rho) = \tau$, with $\tau = 0.1$ in all experiments. For practical compression design, we further introduce two operational sub-regimes within this locally reliable neighborhood, denoted N_1 and N_2:

► **First-order regime** N_1, in which $\left|\frac{1}{2}\delta^\top H(w)\delta\right| \ll |\nabla L(w)^\top \delta|$ and the higher-order remainder is negligible.

► **Second-order regime** N_2, in which the quadratic term is no longer negligible, while the local approximation remains valid and the first-order term remains the primary directional signal.

Candidate-level regime assignment. Based on the layer-wise perturbation sweep, each compression candidate is assigned to the N_1 regime if its loss change is already matched by the first-order approximation within the prescribed tolerance; otherwise, if the same criterion is satisfied only after including the second-order term, it is assigned to the N_2 regime. We emphasize that this is an operational partition used for candidate scoring, rather than a universal theoretical partition of compression regimes. Figure 2 visualizes the resulting empirical regime transitions.

This operational decomposition motivates the following analysis of ordinary and extreme compression regimes.

3.2 Perturbation Neighborhood Decomposition

Building upon the above neighborhood decomposition, we distinguish two regimes for direction-aware compression. In N_1, the first-order term dominates the loss shift, making perturbation direction the primary factor. In N_2, curvature becomes non-negligible and must be incorporated, although the first-order term still provides the main directional signal. These two regimes roughly correspond to ordinary and extreme compression settings, respectively.

3.2.1 Case 1: In the first-order neighborhood N_1. The second-order term is much smaller than the first-order term. This follows from the local nature of the Taylor expansion: when the perturbation magnitude $\|\delta\|$ is sufficiently small compared to the local curvature scale, the quadratic term scales as $O(\|\delta\|^2)$ while the linear term scales as $O(\|\delta\|)$, making the former negligible. This regime corresponds to ordinary compression, such as moderate-rank decomposition or 4/8-bit quantization, where the perturbation magnitude remains relatively small. In this case, if the perturbation is chosen such that $\nabla L(w)^\top \delta < 0$ and higher-order remainder is sufficiently controlled, the approximation predicts a decrease in loss.

3.2.2 Case 2: In the second-order neighborhood N_2. For the N_2, which applies to extreme compression scenarios such as aggressive 2-bit quantization or very low-rank approximations, the directional effects are analyzed in the following two steps.

First-order direction: Gradient-opposing alignment ($\nabla L^\top \delta < 0$) is enforced, making the linear term strictly negative (denoted as $-a$ with $a > 0$). This remains the primary mechanism.

Second-order direction: The quadratic term $b = \frac{1}{2}\delta^\top \mathbf{H}\delta$ behaves in two distinct ways depending on the local Hessian geometry.

◇ Positive contribution case: when the local Hessian is positive semi-definite (convex loss landscape), so $b > 0$. The overall loss change becomes $\Delta L \approx -a + b$. To keep $\Delta L < 0$, the magnitude of the first-order negative contribution a must exceed the second-order positive contribution b (i.e., $a > b$).

◇ Negative curvature case: When $b < 0$, curvature further reduces the local loss shift and therefore makes the perturbation more favorable. In this case, the first-order favorable direction is additionally supported by the local curvature, so the perturbation becomes beneficial in both its linear and quadratic effects.

To validate the above theoretical decomposition and obtain empirical boundaries between N_1 and N_2, we perform controlled perturbation experiments on a small calibration set (typically 128–256 samples). For each target layer, we first compute the gradient ∇L, and then generate perturbations along a chosen direction u with gradually increasing magnitude in the chosen norm. For each perturbation level, we measure the exact loss change ΔL and decompose it using Eq. 3 into the first-order term and the second-order Hessian contribution. This procedure is repeated across multiple layers, perturbation directions, and calibration samples for robustness.

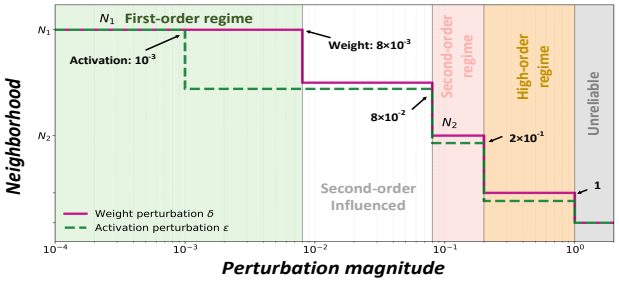

Figure 2: Empirically estimated perturbation regimes under controlled layer-wise perturbations. Solid and dashed step curves denote the transitions for weight perturbations δ and activation perturbations ϵ, respectively.

As shown in Fig. 2, the shaded regions indicate the first-order neighborhood N_1, the second-order neighborhood N_2, a high-order regime, and an unreliable regime as the perturbation magnitude increases, while the intermediate white region marks the transition interval between the first- and second-order neighborhoods.

Empirically, N_1 extends up to approximately 8×10^{-3} for weight perturbations and 10^{-3} for activation perturbations, where the local loss shift is primarily governed by the first-order term. As the perturbation becomes larger, the approximation passes through the transition interval and enters N_2, beginning around 8×10^{-2} and extending to approximately 2×10^{-1}, where curvature effects become non-negligible. Beyond this range, higher-order effects increasingly dominate, and the approximation becomes unreliable near the empirical boundary around 1. These results support the neighborhood decomposition.

These empirical transitions suggest that compression is more predictable when perturbations are evaluated within the appropriate local regime and favored to have a more beneficial loss effect.

> **Main Insight:** Compression-induced perturbations affect loss through both their magnitude and their direction. Within a bounded perturbation neighborhood where the local loss expansion remains reliable, DiRa favors perturbations whose first-order effect is more aligned with reducing the loss, i.e., whose inner product with the loss gradient is more favorable. In moderate compression regimes, a first-order proxy is often sufficient; under stronger compression, DiRa additionally accounts for curvature through a second-order term. This direction-aware local criterion provides a unified principle for both decomposition and mixed-precision quantization.

3.3 A Probabilistic Interpretation of Stability

We further provide a probabilistic interpretation of why direction-aware compression tends to yield more stable outcomes in practice.

The goal is to relate the first-order directional effect to the probability of harmful compression outcomes.

Let z denote a local compression-induced perturbation and let g denote the corresponding local sensitivity vector. Under the first-order approximation, the loss shift satisfies $\Delta L \approx g^\top z$. Viewing $g^\top z$ as a random variable over the space of feasible compression candidates, we summarize its distribution by its mean and variance:

$$\mu = \mathbb{E}[g^\top z], \qquad \sigma^2 = \mathrm{Var}(g^\top z). \tag{5}$$

When candidate selection is directionally favorable on average, we expect $\mu < 0$, meaning that the first-order effect tends to reduce rather than increase the local loss. In this case, degradation corresponds to the event $g^\top z \geq 0$. Applying Cantelli's inequality (the one-sided Chebyshev inequality), we obtain the upper bound

$$\mathbb{P}(g^\top z \geq 0) \leq \frac{\sigma^2}{\sigma^2 + \mu^2}. \tag{6}$$

This expression highlights the desired stability condition: the probability of degradation becomes smaller when the favorable first-order effect dominates its variability, i.e., when $|\mu|^2 \gg \sigma^2$. From this perspective, DiRa improves stability by biasing candidate selection toward perturbations with more favorable directional alignment, thereby making μ more negative and reducing the likelihood of harmful compression outcomes.

4 Instantiations of DiRa

DiRa follows a common offline procedure across compression modes: it estimates local statistics from a small calibration set, determines layer-wise admissible neighborhoods, constructs feasible candidates with explicit directional control, and selects favorable candidates under the target resource budget. In the second-order regime, it further incorporates lightweight curvature information, while all candidate construction and scoring remain calibration-only and do not update model parameters. We next instantiate this procedure for mix quantization (Sec 4.1) and tensor decomposition (Sec 4.2).

4.1 Quantization under DiRa

For mixed-precision quantization, DiRa decomposes the compression decision into two coupled components: (i) *bit-width allocation*, which determines the feasible quantization grid and storage cost, and (ii) *directional rounding*, which explicitly controls the sign and local geometry of the induced perturbation within that grid. This view allows DiRa to treat perturbation direction not as a passive byproduct of quantization, but as an optimization variable constrained by the deployment-compatible quantizer. The same directional-rounding rule is applied to both weights and activations in mixed W/A settings.

For layer i, assigning bit-width j determines a quantization grid $\mathcal{Q}_{i,j}$ and storage cost $S[i][j]$. Under conventional post-training quantization, the quantized value is fixed once the grid is chosen, and the resulting perturbation is treated as immutable. In contrast, DiRa performs *directional rounding* within the feasible grid: for each weight element, it selects the rounding destination that yields the most favorable local loss surrogate while remaining compatible with the underlying quantization scheme.

Since DiRa only chooses between valid neighboring quantization levels already present in $\mathcal{Q}_{i,j}$, the resulting directional quantizer remains fully compatible with the original deployment format.

Specifically, for each scalar weight w_k in layer i, let q_k^- and q_k^+ denote the two neighboring quantization levels in $\mathcal{Q}_{i,j}$. These two choices induce two feasible local perturbations $\delta_k^- = q_k^- - w_k$, $\delta_k^+ = q_k^+ - w_k$. DiRa selects the rounding destination by minimizing a unified local loss surrogate:

$$q_k^\star = \arg\min_{q \in \{q_k^-, q_k^+\}} g_k(q - w_k) + \mathbb{1}_{N_2} \cdot \frac{1}{2} h_k (q - w_k)^2, \tag{7}$$

where g_k is the gradient of the calibration loss with respect to w_k, h_k is a curvature proxy, and $\mathbb{1}_{N_2}$ activates the quadratic term only in the second-order regime N_2. The same directional-rounding rule is applied to activation quantization: for each activation value, DiRa selects between the two valid neighboring activation levels on the feasible activation grid using the analogous first-/second-order local loss surrogate. Therefore, in the first-order regime N_1, DiRa reduces to gradient-guided directional rounding, while in N_2 it incorporates curvature and becomes more conservative. This produces a direction-aware perturbation $\delta_{i,j}^{\mathrm{DiRa}} = Q_{i,j}^{\mathrm{DiRa}}(W_i) - W_i$, for layer i under bit-width j, where $Q_{i,j}^{\mathrm{DiRa}}(\cdot)$ denotes the DiRa directional quantizer.

Algorithm 1 Mixed-Prec Search via Grouped Knapsack under DiRa

Input: Network M, bit-width set $\{q_j\}_{j=1}^K$, capacity C
Output: Bit-width assignment $\{j_i^\star\}_{i=1}^n$

```
Calibrate M to estimate local gradients and curvature proxies
for each layer i and bit-width j do
    Compute storage cost S[i][j]
    Construct the feasible quantization grid Q_{i,j}
    Perform directional rounding within Q_{i,j} to obtain δ^{DiRa}_{i,j}
    if candidate (i, j) is scored in N_1 then
        Compute P[i][j] = ΔL̂_1(δ^{DiRa}_{i,j})
    else
        Compute P[i][j] = ΔL̂_2(δ^{DiRa}_{i,j})
    end if
end for
Solve grouped knapsack under capacity C
return bit-width assignment
```

The candidate associated with assigning bit-width j to layer i is then computed from the resulting direction-controlled perturbation.

$$P[i][j] = \nabla L(w)^\top \delta_{i,j}^{\mathrm{DiRa}} + \mathbb{1}_{N_2} \cdot \frac{1}{2} (\delta_{i,j}^{\mathrm{DiRa}})^\top H(w)\, \delta_{i,j}^{\mathrm{DiRa}}. \tag{8}$$

Each layer forms a group, each candidate bit-width is an item, and the storage budget defines the knapsack capacity. The mixed-precision optimization is formulated as

$$\begin{aligned} \min_{\{j_i\}} \quad & \sum_{i=1}^n P[i][j_i] \\ \text{s.t.} \quad & \sum_{i=1}^n S[i][j_i] \leq C, \quad j_i \in \{1, \ldots, K\}, i = 1, \ldots, n. \end{aligned} \tag{9}$$

In DiRa, candidates are scored after directional rounding within the feasible quantization set. Bit-width allocation determines *where* compression is applied, while rounding direction determines *how* it affects local loss.

Implementation details. Each bit-width candidate follows the quantizer configuration of the target deployment backend. In mixed

W/A quantization, weight and activation candidates are coupled through each layer's deployment bit-width configuration, which determines the feasible grids, storage cost, and candidate score. DiRa does not modify the quantization grid, but only the rounding decision between the two valid neighboring levels. Candidates are assigned to N_1 or N_2 according to whether their induced perturbations fall within the corresponding layer-wise admissible neighborhood estimated in Section 3, and are then selected by grouped knapsack. In N_2, h_k is implemented as a diagonal Hessian proxy [9] from calibration-time Hessian-vector products and used only for offline scoring. Follow [25, 31], activation-side directional information is used only during calibration to determine deployment-compatible quantization parameters, so inference applies the exported static quantizer directly. This keeps DiRa fully compatible with the original mixed-precision deployment format, without introducing extra quantization states or runtime operators.

4.2 Decomposition under DiRa

For tensor decomposition, DiRa views compression as a constrained perturbation design problem. Given a target rank c, the low-rank parameterization defines a locally feasible perturbation family, and DiRa biases the perturbation toward a gradient-opposing direction within that set, rather than relying only on reconstruction fidelity.

Let $\hat{W}_{\text{svd}}^{(c)}$ denote the standard rank-c approximation of layer weight W. Its induced perturbation is $\delta_{\text{svd}}^{(c)} = \hat{W}_{\text{svd}}^{(c)} - W$.

Since the direction of $\delta_{\text{svd}}^{(c)}$ is determined purely by reconstruction, it is not necessarily favorable to the local loss. To introduce explicit directional control, DiRa applies a gradient-guided correction inside the feasible rank-c subspace $\mathcal{S}_c$:

$$\tilde{\delta}^{(c)} = \Pi_{\mathcal{S}_c}\left(\delta_{\text{svd}}^{(c)} - \alpha G\right), \qquad G = \nabla_W L, \tag{10}$$

where $\mathcal{S}_c$ denotes the local feasible subspace induced by the retained rank-c singular directions, $\alpha > 0$ is a steering coefficient, and $\Pi_{\mathcal{S}_c}$ denotes projection onto the feasible low-rank subspace. In this way, the rank constraint determines the admissible perturbation family, while the gradient term biases the realized perturbation toward local descent.

The direction-corrected perturbation is then evaluated under the neighborhood-aware local surrogate:

$$\widehat{\Delta L}(\tilde{\delta}^{(c)}) = \nabla L(w)^\top \tilde{\delta}^{(c)} + \mathbb{1}_{N_2} \cdot \frac{1}{2}(\tilde{\delta}^{(c)})^\top H(w)\tilde{\delta}^{(c)}, \tag{11}$$

where the quadratic term is activated only in the second-order regime N_2. DiRa finally selects the smallest feasible rank whose corrected perturbation remains inside the admissible neighborhood and yields a favorable projected loss.

Implementation details. The shared calibration, neighborhood estimation, and candidate filtering procedure follows the common pipeline described at the beginning of this section. Let $W \approx U_c \Sigma_c V_c^\top$ be the truncated SVD of rank c. We define the feasible correction subspace using the retained singular directions and implement the projection as $\Pi_{\mathcal{S}_c}(X) = U_c U_c^\top X V_c V_c^\top$, or an equivalent projection onto the bilinear span induced by (U_c, V_c). This ensures that the corrected perturbation remains representable within the same target rank. In practice, the steering coefficient α is selected from a small validation grid and then fixed for all layers of the same model.

Algorithm 2 Layer-wise Rank Selection under DiRa

Input: Layer weight W, maximum rank $r_{\max}$, neighborhood constraint $N_{\rho,\eta}$, steering coefficient α
Output: Selected rank $c^\star$

```
Estimate layer gradient G
for c = 1 to r_max do
    Construct rank-c approximation Ŵ_svd^(c)
    Compute δ_svd^(c) = Ŵ_svd^(c) − W
    Compute corrected perturbation δ̃^(c) = Π_{S_c}(δ_svd^(c) − αG)
    if δ̃^(c) ∉ N_{ρ,η} then
        continue
    end if
    Compute projected loss surrogate ΔL̂(δ̃^(c))
    Record candidate score
end for
Select the smallest feasible rank with favorable projected loss
return c*
```

After the rank is selected, we re-factorize the corrected weight matrix $W + \tilde{\delta}^{(c)}$ into a rank-c form for deployment, introducing no additional runtime cost beyond standard low-rank inference.

5 Experiments

We organize the experiments around 4 research questions (RQ):

- **RQ1: How effective is DiRa for mix quantization?**
- **RQ2: How effective is DiRa for tensor decomposition?**
- **RQ3: How does DiRa perform in practical deployment?**
- **RQ4: How is the design of the DiRa framework validated?**

5.1 Experimental Setup

Datasets. We evaluate DiRa on both vision and language tasks. For image classification, we use ImageNet-1K [19]. For natural language understanding, we use SQuAD [29], MNLI [36], and MMLU [11]. For additional LLM evaluation, we further report results on zero-shot commonsense benchmarks including BoolQ [7], PIQA [6], HellaSwag [42], and WinoGrande [30].

Models and metrics. Our evaluation covers CNNs, vision transformers, BERT-style language models, and decoder-only large language models. For image classification, we report Top-1/Top-5 accuracy. For SQuAD, we report EM/F1. For MNLI, MMLU, and other LLM benchmarks, we report accuracy. For deployments, we additionally report memory, latency.

Implementation environment. All experiments start from publicly available full-precision checkpoints implemented in PyTorch. Unless otherwise stated, experiments are conducted on two NVIDIA A800 GPUs. No retraining is used after compression. In our experiments, we use 128–256 randomly sampled training images for ImageNet-1K, small held-out subsets for SQuAD and MNLI, and 512 WikiText samples for MMLU and other zero-shot LLM benchmarks. For decoder-only LLMs, calibration uses language-model cross-entropy on WikiText, while zero-shot tasks such as MMLU are used only for evaluation. All comparisons are performed under matched compression settings. For decomposition, methods are compared at identical or near-identical compression ratios. For

Table 1: Mixed-precision quantization results of DiRa across CV and NLP tasks. N_1 and N_2 denote the first- and second-order perturbation neighborhoods, respectively. Entropy refers to cross-entropy, and Comp. refers to compression ratio.

Task	Model	Metric	Full-Prec.	DiRa-N_1			DiRa-N_2		
				Result	Entropy ↓	Comp. ↓	Result	Entropy ↓	Comp. ↓
MNIST	CNN	Top-1	97.51	**97.66**	7.92 → 7.86 ±0.019	73%	97.45	7.92 → 7.97 ±0.010	**92%**
CIFAR	VGG13	Acc.	73.69	**74.09**	127.26 → 125.03 ±0.062	74%	72.67	127.26 → 129.00 ±0.044	**91%**
	MobileNet_V2	Acc.	62.44	**62.88**	163.58 → 162.45 ±0.023	71%	**62.45**	163.58 → 163.58 ±0.070	**86%**
ImageNet	VGG16_BN	Top-1/Top-5	73.34 / 91.51	**73.71 / 91.52**	106.62 → 105.43 ±0.009	66%	71.97 / 90.11	106.62 → 108.02 ±0.014	**90%**
	MobileNet_V1	Top-1/Top-5	70.28 / 89.43	**70.84 / 89.68**	114.79 → 114.66 ±0.014	68%	69.34 / 87.91	114.79 → 116.06 ±0.009	**85%**
	MobileNet_V2	Top-1/Top-5	71.89 / 90.29	**71.89 / 90.30**	114.80 → 114.78 ±0.003	71%	70.01 / 88.97	114.80 → 116.10 ±0.002	**83%**
	ResNet-50	Top-1/Top-5	76.40 / 93.10	**76.44 / 93.12**	100.19 → 98.54 ±0.082	66%	76.25 / 92.68	100.19 → 101.93 ±0.066	**87%**
SQuAD	BERT	EM / F1	80.49 / 88.15	**80.51 / 88.15**	44.61 → 44.61 ±0.002	45%	79.12 / 86.99	44.61 → 45.70 ±0.004	**74%**
LLM	TinyLlama	Acc.	26.93	**27.01**	-	22%	25.93	-	**71%**

quantization, we match candidate bit-width ranges and use the same evaluation protocol. For LLM benchmarks, all methods are evaluated under the same prompting and zero-shot settings.

Table 2: Comparison of 4-bit (W4A4) post-training quantization results on LLaMA2-13B, Qwen3-8B and LLaMA3-8B across zero-shot commonsense benchmarks.

Models	Methods	BoolQ	PIQA	HellaSwag	WinoGrande
LLaMA2-13B	FP16	80.5	80.7	79.4	72.1
	QUIK [2]	71.30	75.70	73.30	64.60
	QuaRot [3]	76.63	79.07	75.84	69.01
	ResQ [31]	**79.71**	79.14	77.9	69.89
	Ours	79.69	**79.5**	**78.3**	**69.91**
Qwen3-8B	FP16	86.71	76.64	57.10	68.04
	GPTQ [9]	86.30	76.33	55.90	68.50
	FAQ [37]	85.77	**76.80**	55.91	68.20
	Ours	**86.40**	76.73	**56.30**	**68.59**
LLaMA3-8B	FP16	77.74	81.44	79.15	79.16
	ResQ	72.50	78.31	76.50	71.00
	BDialect [16]	77.37	75.24	74.12	66.38
	Ours	**77.40**	**79.92**	**78.51**	**73.01**

5.2 RQ1: Results on Mix Quantization

We first evaluate DiRa on modern large language models, where post-training quantization is particularly challenging. As shown in Table 2, DiRa is competitive with or outperforms recent PTQ baselines on LLaMA2-13B, Qwen3-8B, and LLaMA3-8B across zero-shot commonsense benchmarks. In particular, DiRa achieves the best performance on multiple datasets, demonstrating that the proposed direction-aware criterion transfers effectively to LLM quantization.

We then examine whether this advantage generalizes beyond LLMs. Table 1 summarizes the results across diverse CV and NLP models. In the first-order regime, DiRa-N_1 consistently preserves, and in several cases slightly improves upon, the full-precision baseline, indicating near-lossless mixed-precision quantization under mild perturbations. These results also indicate a more stable compression behavior: under matched compression settings, DiRa preserves performance while often reducing the entropy of the induced compression noise. More importantly, in the more aggressive N_2 regime, DiRa still attains substantially higher compression ratios while maintaining competitive accuracy, showing that the second-order criterion remains effective even under extreme compression.

Table 3: Comparison with representative PTQ baselines under INT2 quantization, where Ours denotes DiRa-N_2.

ResNet18 (INT2) / ImageNet		LLaMA2-7B (INT2) / WikiText	
Method	**Acc. ↑**	**Method**	**PPL ↓**
Full Prec	71.01	Full Prec	5.47
AdaRound	0.11	QuaRot-RTN	1.1e4
BRECQ	62.69	OmniQuant	21.85
Ours	**65.33**	**Ours**	**19.24**

We further compare DiRa-N_2 with representative PTQ methods in Table 3 that explicitly optimize rounding or reconstruction surrogates. While these methods can largely be viewed as Hessian-based surrogates for quantization cost, DiRa uses the local-loss surrogate not merely to estimate perturbation sensitivity, but to explicitly choose the realized perturbation direction among feasible compression candidates. Under aggressive low-bit settings, DiRa remains stronger on both vision and language models: for ResNet18 on ImageNet with W2A4 quantization, it outperforms AdaRound and BRECQ, and for LLaMA2-7B on WikiText with weight-only INT2 quantization, it also improves over QuaRot-RTN [3] and OmniQuant [32]. This suggests that the gain does not come from local sensitivity information alone, but from directly modeling local loss shift and using it to select favorable perturbation directions.

This trend is consistent across CNNs, ImageNet backbones, BERT, and TinyLlama, suggesting that the proposed quantization criterion is broadly applicable rather than architecture-specific. Moreover, the entropy trend is broadly aligned with the accuracy trend, further supporting the validity of the neighborhood-aware loss surrogate. These results show that DiRa provides an effective unified criterion for mixed-precision quantization, delivering stable post-training behavior across both ordinary and extreme compression regimes.

5.3 RQ2: Results on Tensor Decomposition

We first evaluate DiRa on large language models, where post-training decomposition is particularly challenging. As shown in Table 4, DiRa outperforms representative baselines on LLaMA2-7B/13B at a 30% compression ratio, achieving the best results on

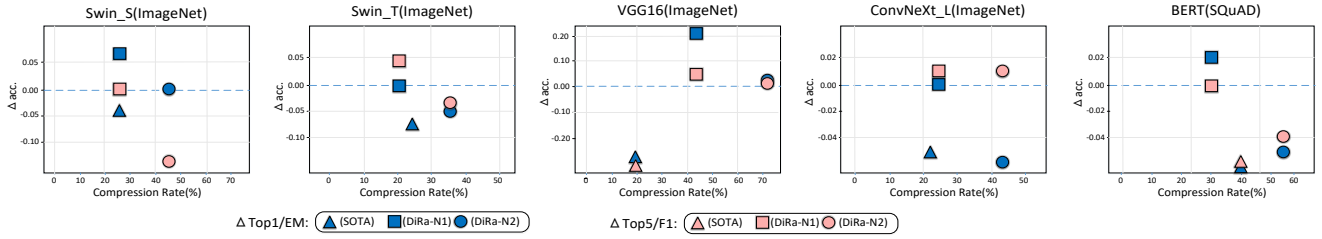

Figure 3: Cross-architecture gains of DiRa. On Swin-Transformer, VGG, ConvNeXt, and BERT, both DiRa-N_1 and DiRa-N_2 consistently outperform prior post-training decomposition methods[10, 18, 41, 45], showing that the proposed direction-aware criterion transfers well across architectures. In several settings, the compressed model even slightly exceeds the original model.

Table 4: Performance comparison of low-rank decomposition methods on LLaMA2 under identical compression ratios.

Method	MMLU	BoolQ	PIQA	WinoGrande
LLaMA2-7B	41.3	77.8	78.1	69.4
LLM-Pruner [27]	26.2	63.7	**75.9**	63.4
FLAP [1]	31.9	53.9	74.5	62.9
SVD-LLM [34]	26.8	54.7	65.1	62.4
SoLA [15]	34.1	75.4	74.6	66.4
Ours	**35.2**	**75.7**	**75.9**	**66.9**
LLaMA2-13B	55.4	80.6	80.4	72.5
SoLA [15]	39.4	77.1	72.6	67.4
Ours	**40.0**	**77.5**	**73.9**	**68.2**

MMLU and WinoGrande while remaining competitive on BoolQ and PIQA. These results verify the effectiveness of the proposed direction-aware criterion in a challenging LLM setting.

We then examine whether this advantage generalizes beyond LLMs. Figure 3 reports the accuracy change (Δacc, defined as compressed minus full-precision accuracy) across diverse mainstream architectures, including Swin-Transformer, VGG, ConvNeXt, and BERT. DiRa generalizes consistently well across these model families, with both DiRa-N_1 and DiRa-N_2 outperforming existing post-training decomposition baselines. The gains in Figure 3 are also more consistent across architectures, suggesting that the effect of DiRa is not limited to a specific model family or compression point.

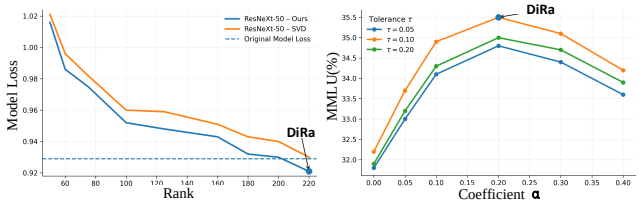

Figure 4: (Left) Decomposition behavior under rank. (Right) tolerance τ, and correction strength α.

More notably, DiRa-N_1 yields positive Δacc in several settings, meaning that the compressed model even slightly surpasses the full-precision baseline. Such behavior is rarely observed in prior post-training decomposition methods. We attribute this to the direction-aware nature of DiRa: when the induced perturbation remains within N_1, a favorable perturbation direction can produce a beneficial first-order effect, so decomposition does not necessarily act as purely harmful noise. In the more aggressive N_2 regime, DiRa still maintains a stronger accuracy–compression trade-off than prior methods, supporting the usefulness of the second-order criterion under larger perturbations.

Figure 4 (left) further illustrates why direction-aware evaluation matters in tensor decomposition. On ResNext-50, DiRa and standard SVD show clearly different local loss trends under similar rank choices, suggesting that decomposition quality is not determined by compression level alone. Unlike SVD, which mainly minimizes reconstruction error and ignores the perturbation direction, DiRa evaluates candidates by their directional effect on loss, and therefore identifies better decomposition candidates with lower loss.

5.4 RQ3: Practical Deployment

Beyond benchmark accuracy, we measure LLaMA-7B end-to-end inference latency and memory usage on an NVIDIA A800 GPU using the FasterTransformer implementation (fixed input length 15, batch size 1, output lengths 128 and 256). DiRa-N1 (W8A8) and DiRa-N2 (2/4-bit) correspond to the first-order and second-order perturbation neighborhoods, respectively.

Table 5: Inference latency and memory usage of the FasterTransformer implementation on a single NVIDIA A800 GPU.

Method	Output Length = 128		Output Length = 256	
	Latency	Memory	Latency	Memory
FP16 Baseline	1497.4 ms	13.47 GB	3239.8 ms	13.68 GB
CUTLASS(W8A16)	868.4 ms	7.39 GB	1800.4 ms	7.46 GB
SmoothQuant (W8A8)	841.8 ms	7.39 GB	1714.8 ms	7.46 GB
DiRa-N_1 (W8A8)	833.1 ms	7.39 GB	1688.2 ms	7.46 GB
DiRa-N_2 (2/4-bit)	**751.0** ms	**4.79** GB	**1400.1** ms	**4.80** GB

Table 5 reports the results. In N_1, DiRa-N_1 reduces memory by about 45% and latency by 44–48% relative to FP16. In the aggressive N_2 regime, memory further drops to 4.79–4.80 GB (64% lower than FP16), while latency reaches 751 ms and 1400 ms for output lengths 128 and 256. Compared with SmoothQuant and CUTLASS, DiRa consistently achieves the best memory–latency trade-off, enabling efficient single-node LLM deployment on commodity GPUs.

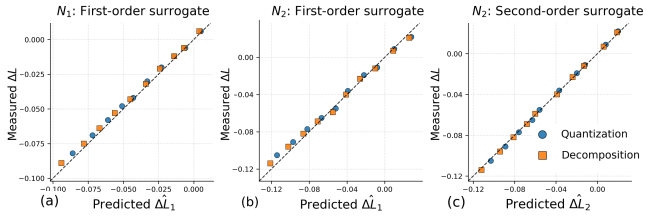

Figure 5: Predicted versus measured loss shift in N_1 and N_2. The first-order surrogate is accurate in N_1, while the second-order surrogate is more accurate in N_2.

5.5 RQ4: Ablation Studies

Validity of Neighborhood-Aware Surrogates. Figure 5 directly validates the local surrogates used by DiRa for candidate scoring. In the first-order regime N_1, the predicted loss shift from the first-order surrogate is already well aligned with the measured loss shift, indicating that first-order information is sufficient for reliable candidate evaluation in the small-perturbation regime. In contrast, in the stronger-perturbation regime N_2, the first-order surrogate becomes noticeably less accurate, as reflected by the larger deviation of the points from the ideal diagonal line. Once the curvature term is included, however, the second-order surrogate becomes much more closely aligned with the measured loss shift. This result directly supports the regime-dependent design of DiRa: first-order scoring is appropriate in N_1, while second-order scoring is necessary in N_2 to better capture the loss behavior of compression candidates. In Appendix, we also vary the approximation tolerance τ used for neighborhood assignment and observe only minor changes in final performance, suggesting that the induced N_1/N_2 partition is reasonably stable around the default setting.

Table 6: Ablation study on noise amplitude and direction control across different quantization levels (INT2/4/8).

Method	Noise		Top-1(%) ↑	Entropy ↓
	Amplitude	Direction		
FP32 Baseline	×	×	73.69	1.28
INT8	✓	×	69.32	1.92
INT4/8	✓	×	66.32	2.62
DiRa INT8	✓	Random	68.94	1.83
DiRa INT8	✓	✓	**73.70**	1.21
DiRa INT4/8	✓	✓	71.94	**1.19**
DiRa INT2/4/8	✓	✓	66.58	2.59

Table 7: Calibration set size sensitivity of DiRa for mixed-precision quantization and tensor decomposition.

Compression	Model	Metric	16	128	256	512
Quantization	ResNet50	Top-1 (%)	70.62	75.97	**76.44**	76.43
	Qwen3-8B	MMLU (%)	66.99	69.64	70.04	**71.01**
Decomposition	ResNet50	Top-1 (%)	67.43	70.28	72.34	**74.04**
	LLaMA2-7B	MMLU (%)	24.30	30.34	34.60	**35.22**

Direction Control and Calibration Size. We further isolate the role of perturbation magnitude and perturbation direction in Table 6. Magnitude-only quantization degrades substantially as compression becomes more aggressive. A random directional choice does not recover this loss, showing that the benefit is not due to directional freedom alone. In contrast, DiRa significantly restores performance under the same quantization settings, supporting our claim that stable compression depends not only on perturbation magnitude but also on the directional effect of the induced perturbation. Since all variants use the same candidate bit-width settings, the gain mainly comes from direction-aware candidate selection rather than a looser compression budget. Finally, Table 7 studies the sensitivity of DiRa to calibration set size in both quantization and tensor decomposition. Performance improves markedly from very small to moderate calibration sets and then gradually saturates, suggesting that useful local loss estimates can be obtained with relatively few samples.

Table 8: Runtime comparison with representative quantization methods in ImageNet/WikiText-2.

4-bit	Method	Acc/PPL	Runtime	#GPU
ResNet18	PD-Quant	69.23	62.0 min	**1**
	DiRa-N_2	**69.31**	**11.2 min**	**1**
LLaMA2-7B	SpinQuant	5.6	95.0 min	4
	DiRa-N_2	5.6	**33.3 min**	**1**

Runtime. Under similar settings, Table 8 summarizes the practical runtime of DiRa under 4-bit settings, together with the corresponding GPU usage. DiRa achieved the lowest runtime and the fewest GPUs used. In practice, the additional calibration overhead remains modest; for instance, first-order(N_1) calibration on ResNet-50 takes only **0.9** minutes, second-order (N_2) takes **8.8** minutes. Detailed calibration-time results are deferred to Appendix.

Sensitivity to τ and α. We further study the decomposition instantiation on LLaMA2-7B by varying the tolerance τ and the correction strength α. As shown in Figure 4 (right), the performance consistently improves as α increases from zero, reaches its best region around the DiRa setting, and then gradually declines when the correction becomes too strong. Across different τ values, the overall trend remains similar and the performance variation is moderate, suggesting that the decomposition variant is not overly sensitive to either hyperparameter within a reasonable range.

Additional details are provided in ***Appendix***, including results on larger LLMs, an analysis of cross-layer interactions, N_1/N_2 regime assignment details and results for Hessian-proxy estimation.

6 Conclusion

We presented DiRa, a training-free and model-agnostic framework for post-training compression. By introducing neighborhood-aware local loss surrogates, DiRa provides a unified direction-aware criterion for both tensor decomposition and mixed-precision quantization. Experiments across vision and language models show that this principle leads to more stable compression behavior and strong accuracy–compression trade-offs. These findings suggest that reliable post-training compression depends not only on perturbation magnitude, but also on perturbation direction.

References

[1] Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. 2024. Fluctuation-based adaptive structured pruning for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 10865–10873.

[2] Saleh Ashkboos, Ilia Markov, Elias Frantar, Tingxuan Zhong, Xincheng Wang, Jie Ren, Torsten Hoefler, and Dan Alistarh. 2023. QUIK: Towards End-to-End 4-Bit Inference on Generative Large Language Models. arXiv:2310.09259 [cs.LG] https://arxiv.org/abs/2310.09259

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

[4] Ron Banner, Yury Nahshan, Elad Hoffer, and Daniel Soudry. 2018. Aciq: Analytical clipping for integer quantization of neural networks. (2018).

[5] James Bisgard. 2020. *Analysis and linear algebra: the singular value decomposition and applications*. Vol. 94. American Mathematical Soc.

[6] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. PIQA: Reasoning about Physical Commonsense in Natural Language. arXiv:1911.11641 [cs.CL] https://arxiv.org/abs/1911.11641

[7] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. arXiv:1905.10044 [cs.CL] https://arxiv.org/abs/1905.10044

[8] Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2020. Hawq-v2: Hessian aware trace-weighted quantization of neural networks. *Advances in neural information processing systems* 33 (2020), 18518–18529.

[9] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. arXiv:2210.17323 [cs.LG] https://arxiv.org/abs/2210.17323

[10] Shangqian Gao, Ting Hua, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. 2024. Adaptive Rank Selections for Low-Rank Approximation of Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 227–241. doi:10.18653/v1/2024.naacl-long.13

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

[12] Samuel Horváth, Stefanos Laskaridis, Shashank Rajput, and Hongyi Wang. 2024. Maestro: uncovering low-rank structures via trainable decomposition. In *Proceedings of the 41st International Conference on Machine Learning* (Vienna, Austria) *(ICML'24)*. JMLR.org, Article 758, 23 pages.

[13] Jie Hu, Mengze Zeng, and Enhua Wu. 2023. Bag of Tricks with Quantized Convolutional Neural Networks for image classification. arXiv:2303.07080 [cs.CV] https://arxiv.org/abs/2303.07080

[14] Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and Xiaojuan Qi. 2025. SliM-LLM: Salience-Driven Mixed-Precision Quantization for Large Language Models. arXiv:2405.14917 [cs.LG] https://arxiv.org/abs/2405.14917

[15] Xinhao Huang, You-Liang Huang, and Zeyi Wen. 2025. SoLA: Leveraging Soft Activation Sparsity and Low-Rank Decomposition for Large Language Model Compression. *Proceedings of the AAAI Conference on Artificial Intelligence* 39, 16 (Apr. 2025), 17494–17502. doi:10.1609/aaai.v39i16.33923

[16] Wonsuk Jang and Thierry Tambe. 2025. BlockDialect: Block-wise Fine-grained Mixed Format Quantization for Energy-Efficient LLM Inference. arXiv:2501.01144 [cs.CL] https://arxiv.org/abs/2501.01144

[17] Alper Kalle, Theo Rudkiewicz, Mohamed-Oumar Ouerfelli, and Mohamed Tamaazousti. 2025. Distribution-Aware Tensor Decomposition for Compression of Convolutional Neural Networks. arXiv:2511.04494 [cs.LG] https://arxiv.org/abs/2511.04494

[18] M. Kokhazadeh, G. Keramidas, and V. Kelefouras. 2025. Efficient CNN Compression via Multi-method Low Rank Factorization and Feature Map Similarity. arXiv:2510.00062 [cs.CV] https://arxiv.org/abs/2510.00062

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2017. ImageNet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (2017), 84–90.

[20] Nan Li, Yonghui Su, and Lianbo Ma. 2025. Efficient and Generalizable Mixed-Precision Quantization via Topological Entropy. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[21] Xin Li, Shuai Zhang, Bolan Jiang, Yingyong Qi, Mooi Choo Chuah, and Ning Bi. 2019. DAC: Data-Free Automatic Acceleration of Convolutional Networks. 1598–1606. doi:10.1109/WACV.2019.00175

[22] 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. *arXiv preprint arXiv:2102.05426* (2021).

[23] 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 LLM Compression and Acceleration. arXiv:2306.00978 [cs.CL] https://arxiv.org/abs/2306.00978

[24] Xingchao Liu, Mao Ye, Dengyong Zhou, and Qiang Liu. 2021. Post-training Quantization with Multiple Points: Mixed Precision without Mixed Precision. arXiv:2002.09049 [cs.LG] https://arxiv.org/abs/2002.09049

[25] Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. 2024. Spinquant: Llm quantization with learned rotations. *arXiv preprint arXiv:2405.16406* (2024).

[26] Qian Lou, Feng Guo, Lantao Liu, Minje Kim, and Lei Jiang. 2019. Autoq: Automated kernel-wise neural network quantization. *arXiv preprint arXiv:1902.05690* (2019).

[27] Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. LLM-Pruner: On the Structural Pruning of Large Language Models. arXiv:2305.11627 [cs.CL] https://arxiv.org/abs/2305.11627

[28] Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. 2020. Up or Down? Adaptive Rounding for Post-Training Quantization. arXiv:2004.10568 [cs.LG] https://arxiv.org/abs/2004.10568

[29] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* (2016).

[30] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. arXiv:1907.10641 [cs.CL] https://arxiv.org/abs/1907.10641

[31] Utkarsh Saxena, Sayeh Sharify, Kaushik Roy, and Xin Wang. 2025. ResQ: Mixed-Precision Quantization of Large Language Models with Low-Rank Residuals. arXiv:2412.14363 [cs.LG] https://arxiv.org/abs/2412.14363

[32] Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. 2024. Omni-Quant: Omnidirectionally Calibrated Quantization for Large Language Models. arXiv:2308.13137 [cs.LG] https://arxiv.org/abs/2308.13137

[33] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. 2019. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8612–8620.

[34] Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. 2025. SVD-LLM: Truncation-aware Singular Value Decomposition for Large Language Model Compression. arXiv:2403.07378 [cs.CL] https://arxiv.org/abs/2403.07378

[35] Yutong Wang, Haiyu Wang, and Sai Qian Zhang. 2025. QSVD: Efficient Low-rank Approximation for Unified Query-Key-Value Weight Compression in Low-Precision Vision-Language Models. *arXiv preprint arXiv:2510.16292* (2025).

[36] Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426* (2017).

[37] Haiyang Xiao, Weiqing Li, Jinyue Guo, Guochao Jiang, Guohua Liu, and Yuewei Zhang. 2026. FAQ: Mitigating Quantization Error via Regenerating Calibration Data with Family-Aware Quantization. arXiv:2601.11200 [cs.LG] https://arxiv.org/abs/2601.11200

[38] Yuhui Xu, Yuxi Li, Shuai Zhang, Wei Wen, Botao Wang, Wenrui Dai, Yingyong Qi, Yiran Chen, Weiyao Lin, and Hongkai Xiong. 2019. Trained rank pruning for efficient deep neural networks. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS)*. IEEE, 14–17.

[39] Huanrui Yang, Minxue Tang, Wei Wen, Feng Yan, Daniel Hu, Ang Li, Hai Li, and Yiran Chen. 2020. Learning low-rank deep neural networks via singular vector orthogonality regularization and singular value sparsification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 678–679.

[40] Xiyu Yu, Tongliang Liu, Xinchao Wang, and Dacheng Tao. 2017. On compressing deep models by low rank and sparse decomposition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7370–7379.

[41] Zhihang Yuan, Yuzhang Shang, Yue Song, Dawei Yang, Qiang Wu, Yan Yan, and Guangyu Sun. 2025. ASVD: Activation-aware Singular Value Decomposition for Compressing Large Language Models. arXiv:2312.05821 [cs.CL] https://arxiv.org/abs/2312.05821

[42] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? arXiv:1905.07830 [cs.CL] https://arxiv.org/abs/1905.07830

[43] Boyang Zhang, Daning Cheng, Yunquan Zhang, and Fangmin Liu. 2024. FP=xINT: A Low-Bit Series Expansion Algorithm for Post-Training Quantization. *arXiv preprint arXiv:2412.06865* (2024).

[44] Boyang Zhang, Suping Wu, Leyang Yang, Bin Wang, and Wenlong Lu. 2023. A Lightweight Grouped Low-rank Tensor Approximation Network for 3D Mesh Reconstruction From Videos. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 930–935.

[45] Cem Üyük, Mike Lasby, Mohamed Yassin, Utku Evci, and Yani Ioannou. 2025. Learning Parameter Sharing with Tensor Decompositions and Sparsity.

arXiv:2411.09816 [cs.LG] https://arxiv.org/abs/2411.09816