RESC-Q: RESIDUAL-AWARE RELIABLE PTQ WITH SHRINKAGE-CONTROLLED COMPENSATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training quantization (PTQ) reduces the memory footprint and inference cost of LLMs under tight deployment constraints. Hessian-based PTQ methods such as GPTQ use estimated curvature information for error compensation, but compensation estimated from calibration data does not always lead to optimal reconstruction loss on unseen inputs. This becomes especially problematic in ultra low-bit quantization. We propose ReSC-Q, which controls uncertain cross-group curvature through shrinkage while explicitly accounting for the reconstruction loss remaining after compensation. ReSC-Q supports practical group-wise quantization by separating the blockwise reconstruction objective into cross-group compensation and within-group quantization, applying shrinkage and residual-aware weighting to each, respectively. Across seven LLMs evaluated at 2-bit, ReSC-Q improves average zero-shot accuracy by 1.09 percentage points over the strongest baseline for each model, reducing the mean accuracy gap to full precision by 23.2%.

1 INTRODUCTION

LLMs are increasingly deployed under tight memory and latency constraints, making post-training quantization (PTQ) essential for reducing model size and inference cost. Among PTQ techniques, Hessian-based methods such as GPTQ have been effective: they quantize weights sequentially and compensate for the resulting error using estimated curvature information. However, at low bit-widths, such as 2-bit quantization, the reliability of this estimated curvature becomes critical.

In this work, we examine how the reliability of Hessian estimates affect error compensation. Off-diagonal Hessian entries are less reliable than diagonal entries under limited calibration data, while error compensation directly relies on these correlations to propagate quantization errors to the remaining weights. More importantly, our analysis shows that compensation estimated from calibration data can increase reconstruction loss on unseen inputs, particularly when calibration data are limited. This motivates a more conservative use of the Hessian information for the quantization process.

We propose ReSC-Q, a Hessian-based PTQ designed around this observation. ReSC-Q organizes the reconstruction objective over blocks aligned with practical quantization groups and separates the standard blockwise compensation term from the residual that remains after compensation. It applies cross-group shrinkage to suppress unreliable off-diagonal Hessian terms used for error compensation, while introducing an upper-bound objective for the post-compensation residual for quantization.

This design jointly addresses how quantization error is compensated and how the remaining reconstruction error is accounted for during quantization. The resulting residual-aware objective can be incorporated into different quantization representations, with group-wise affine quantization used as our default implementation. Across LLMs of different scales, including models with 100B or more parameters, ReSC-Q improves 2-bit quantization accuracy by 1.09 percentage points over the strongest baselines, reducing the mean accuracy gap to full precision by 23.2%.

2 RELATED WORK

Early PTQ methods for LLMs focus primarily on activation outliers. SmoothQuant (1) reduces activation quantization difficulty by migrating activation variance into weights. AWQ (2) scales weights using activation statistics, motivated by the observation that values near range-boundaries

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.

tend to incur smaller errors. However, shifting outlier magnitudes into weights can make ultra low-bit quantization difficult. Other approaches isolate outliers or reorganize channels to mitigate precision loss: LLM.int8() (3) and OWQ (4) keep a small set of sensitive channels in higher precision, while other methods such as (5; 6; 7) use channel permutation to better fit low-precision constraints.

Quantization sensitivity information has also been incorporated directly into the quantization objective. SqueezeLLM (8) uses empirical-Fisher sensitivity to learn non-uniform quantization levels, while LeanQuant (9) uses inverse-Hessian-based importance for reconstruction-aware non-uniform quantization. DASH-Q (10) instead uses diagonal Hessian information to weight affine quantization, formulating the quantization objective as weighted linear regression problem.

Another line of research formulates PTQ as reconstruction-error minimization under local curvature. GPTQ (11; 12) performs layer-wise quantization using a second-order approximation and iteratively compensates the error introduced by previously quantized coordinates. QuIP# (13) further extended the process with BlockLDLQ, enabling blockwise error compensation, while QTIP (14) further combines such feedback with trellis-coded quantization. Complementary to these approaches, several methods (12; 15; 16; 17) apply orthogonal transformations to redistribute outlier values across channels to obtain representations that are easier to quantize at low bit-widths. However, even after the transformation, Hessian-based error compensation often remains necessary.

Despite their effectiveness, Hessian-based methods have limitations in ultra low-bit regimes. Recent studies (2; 18) report that full-Hessian optimization with limited calibration data can overfit, resulting in a gap between perplexity and downstream task accuracy. Arai et al. (19) further note that such local approximations can amplify error accumulation across layers.

3 BACKGROUND

Weight-only quantization maps high-precision weights $W \in \mathbf{R}^{d_{out} \times d_{in}}$ to a low-bit representation Q such that the reconstructed weights $\hat{W}$ incur minimal error. A widely used approach is uniform affine quantization, defined as:

$$Q = \text{clip}\left(\left\lfloor \frac{W+z}{s} \right\rceil, 0, 2^b - 1\right), \quad \hat{W} = s \cdot Q - z, \qquad s = \frac{\max(W) - \min(W)}{2^b - 1}, \quad z = -\min(W) \tag{1}$$

where s and z are the scale and zero-point for b-bit quantization. In LLMs, quantization is typically applied at a per-group granularity to reduce quantization error, called *group-wise quantization*.

PTQ commonly determines the quantized weights by minimizing the difference between the original and quantized layer outputs. Given calibration inputs $X \in \mathbb{R}^{d_{\text{in}} \times N}$, the reconstruction objective is

$$\min_{\hat{W}} \|WX - \hat{W}X\|_F^2 = \sum_{i=1}^{d_{\text{out}}} \min_{\hat{W}_{i,:}} \|W_{i,:}X - \hat{W}_{i,:}X\|_2^2. \tag{2}$$

Since this objective is separable across output channels, each row can be quantized independently. Hessian-based methods (20; 11) use a second-order reconstruction objective to determine error compensation. For a single row $w = W_{i,:}$, the reconstruction loss can be written as:

$$\mathcal{L}(\hat{w}) \approx (w - \hat{w})\mathbf{H}(w - \hat{w})^\top, \qquad \mathbf{H} \approx \hat{\mathbf{H}} = XX^\top, \tag{3}$$

where $\mathbf{H}$ is the Hessian matrix capturing second-order information, and $\hat{\mathbf{H}}$ denotes its estimate computed from the calibration data.

Let w_j be the j-th entry of w, and let $w_j - \hat{w}_j$ be the error introduced by quantizing it to $\hat{w}_j$. Let $\mathcal{R}_j$ denote the other unquantized coordinates. The resulting output error is $(w_j - \hat{w}_j)X_{j,:}$, while adjusting the later weights by $\delta_{\mathcal{R}_j}$ changes the output by $\delta_{\mathcal{R}_j} X_{\mathcal{R}_j,:}$. Under the quadratic objective, the minimum local loss introduced by this quantization and the corresponding weight update are

$$\Delta\mathcal{L}_j^\star = \min_{\delta_{\mathcal{R}_j}} \left\|(w_j - \hat{w}_j)X_{j,:} - \delta_{\mathcal{R}_j} X_{\mathcal{R}_j,:}\right\|_2^2 = \frac{(w_j - \hat{w}_j)^2}{(\mathbf{H}^{-1})_{jj}}, \qquad \delta_{\mathcal{R}_j}^\star = -\frac{w_j - \hat{w}_j}{(\mathbf{H}^{-1})_{jj}}(\mathbf{H}^{-1})_{j,\mathcal{R}_j}. \tag{4}$$

Here, w_j includes previous compensation updates, and $\mathbf{H}$ is restricted to the active coordinates that are not quantized yet, with previously quantized weights held fixed. The update quantizes w_j to $\hat{w}_j$ and adjusts the other active weights through the off-diagonal inverse-Hessian entries.

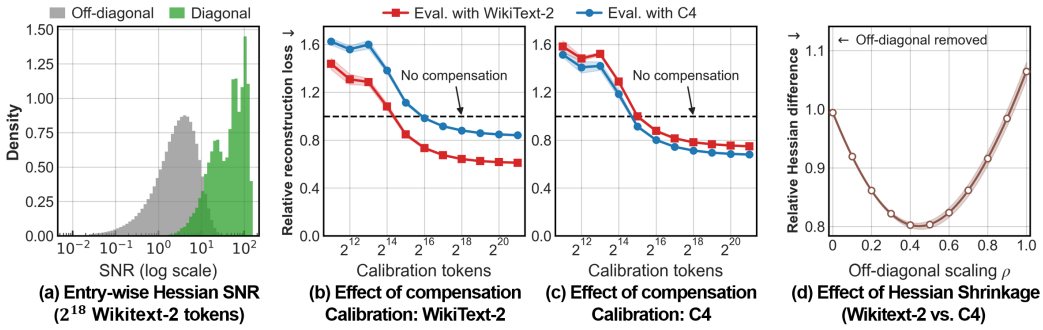

Figure 1: Unreliability of Hessian estimates and its effect on error compensation. (a) Distribution of SNR separately for diagonal and off-diagonal Hessian entries estimated from 2^{18} WikiText-2 tokens. (b, c) Relative reconstruction loss after error compensation using Hessian estimates from WikiText-2 and C4, respectively, with varying calibration sizes. Evaluation uses the samples not used for the calibration, and the reconstruction loss is normalized by the corresponding loss without compensation. (d) Relative ℓ_1 difference between a 2^{18}-token WikiText-2 Hessian estimate and a C4 reference Hessian computed from 2^{24} tokens, under varying off-diagonal shrinkage ρ. Results are averaged over the 5th, 15th, and 25th layers of `Llama-3.1-8B-Instruct`.

The minimum in equation 4 is attained with optimal compensation on the right-hand term, and its value is equal to that defined by the left-hand term. Thus, compensation need not eliminate the entire reconstruction error, and the remaining loss depends on how the weights are quantized. In practice, the Hessian is estimated from limited calibration data, so the compensation itself is also affected by estimation uncertainty. We examine this issue in the following section.

4 Motivation

The statistical reliability of covariance and curvature estimates has been widely studied: sampling noise can make these estimates unreliable when the number of samples is limited (21; 22; 23; 24). In LLM quantization, because off-diagonal Hessian entries can be sensitive to calibration data, diagonal-only quantization has been considered as a simple alternative (10). However, discarding all off-diagonal entries may also remove feature correlations useful for error compensation. In our preliminary analysis, we therefore examine how the reliability of these estimates affects the reconstruction loss after compensation.

To quantify this unreliability, we sample calibration sets from WikiText-2 and compute a Hessian estimate for each set. The calibration set contains 128 samples of 2,048 tokens, giving 2^{18} tokens in total. We compute the mean and standard deviation of each Hessian entry across these estimates and define its signal-to-noise ratio (SNR) as $\mathrm{SNR}_{ij} = |\mathbb{E}(\hat{H}_{ij})|/\mathrm{Std}(\hat{H}_{ij})$. Fig. 1(a) reports the distribution of SNR separately for diagonal and off-diagonal entries. The off-diagonal entries have consistently lower SNR, indicating larger sampling variation relative to their mean magnitude and less reliable estimates of the underlying feature correlations.

We next examine how this estimation uncertainty affects error compensation on unseen data. For each tested group, we fix its quantization error and compute the compensation update using Hessian estimates from WikiText-2 or C4. We then evaluate the updated weight on separate samples from both datasets, with reconstruction loss normalized by the corresponding loss without compensation. Fig. 1(b, c) reports the results using WikiText-2 and C4 for calibration, respectively.

With limited calibration data, error compensation can increase reconstruction loss. While increasing the calibration size reduces this loss, the gains are limited, plateau after a certain point, and depend on the calibration and evaluation datasets. These results highlight the distinction between optimizing compensation on calibration data and achieving optimal reconstruction loss on unseen inputs: an update that minimizes calibration reconstruction loss may not be optimal for other datasets.

We explore a more conservative use of Hessian estimates by attenuating their off-diagonal entries, which capture correlations between input features used for error compensation. Specifically, we

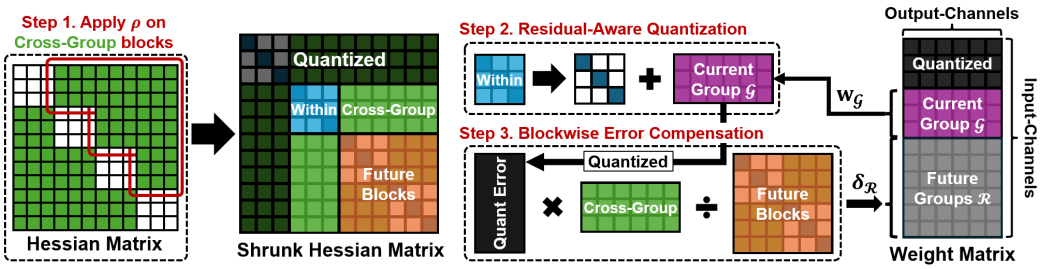

Figure 2: Overview of ReSC-Q. In the initial step, the cross-group blocks of the Hessian are shrunk by a coefficient ρ to suppress unreliable cross-group estimates. Quantization then proceeds group by group: at each step, the current group is quantized using a residual-aware weighted objective, and the resulting quantization error is propagated to future blocks via Group-wise Error Compensation.

consider a *linear shrinkage estimator* that scales the off-diagonal entries while preserving the diagonal ones. Let $\hat{\mathbf{H}} = XX^\top$ denote the Hessian estimate from calibration inputs X, and decompose it into $\mathbf{D} = \mathrm{diag}(\hat{\mathbf{H}})$ and $\mathbf{O} = \hat{\mathbf{H}} - \mathbf{D}$. We define $\tilde{\mathbf{H}}(\rho) = \mathbf{D} + \rho\mathbf{O}$, where $\rho \in [0, 1]$. Thus, $\rho = 0$ removes all off-diagonal entries, while $\rho = 1$ retains the original estimate.

We compare the shrunk Hessian estimated from 2^{18} WikiText-2 tokens against a reference Hessian computed from 2^{24} C4 tokens. As shown in Fig. 1(d), the relative ℓ_1 difference is minimized at an intermediate ρ, rather than at $\rho = 0$ or $\rho = 1$. This suggests that the sampled off-diagonal entries contain useful information, but retaining their full estimated magnitude does not necessarily give the closest agreement with another dataset. Here, the minimum measures agreement with the reference Hessian, not a reconstruction-loss optimum. The result motivates estimating ρ rather than discarding all off-diagonal entries or retaining them without shrinkage.

Our analysis shows that the statistical reliability of Hessian estimates directly affects error compensation. Although off-diagonal entries contain useful feature correlations, compensation optimized on calibration data need not minimize reconstruction loss on unseen inputs. These observations motivate using shrinkage to account for estimation uncertainty while retaining useful correlations, rather than fully trusting or discarding the off-diagonal estimates.

5 Methodology

Hessian-based error compensation (11; 12; 13) relies on unreliable estimates of off-diagonal Hessian to reduce reconstruction loss by sequentially quantizing a group of parameters and propagate the error to the remaining coordinates. However, this mechanism determines only how the error is redistributed; it does not explicitly minimize the residual reconstruction error remaining after compensation.

We address these limitations in ReSC-Q. First, we reorganize the reconstruction objective into compensation blocks aligned with the quantization groups. Second, we isolate the residual loss remaining after compensation and minimize it directly. As shown in Fig. 2, we apply shrinkage ρ to off-diagonal blocks (Step 1) to suppress statistical uncertainty in Hessian estimates. The off-diagonal terms within diagonal blocks are handled separately, as their impact on reconstruction loss differs from off-diagonal blocks.

ReSC-Q quantizes a block of weight parameters at a time. For this step, we design *residual-aware quantization* to further minimize the loss not reduced by error compensation (Step 2 in Fig. 2). After quantizing each block, we compensate for the resulting error using cross-group correlations captured by the off-diagonal blocks and updates the weight parameters in future blocks (Step 3).

We now explain these three steps in detail, starting with Step 3: Blockwise Error Compensation.

5.1 Blockwise Decomposition of Compensation and Residual Loss

Because GPTQ-style error compensation quantizes one coordinate and updates the remaining coordinates, its compensation unit is not directly aligned with practical quantization groups.

BlockLDLQ (13) extends this feedback to jointly quantize and compensate in a blockwise manner. However, this process alone does not determine how the current group should be optimized. We therefore derive a blockwise objective that separates the compensation term from the residual that remains after compensation.

For each output row $\mathbf{w}_i = \mathbf{W}_{i,:} \in \mathbb{R}^{1\times d_{in}}$, let $\{\mathcal{G}_k\}_{k=1}^m$ be the quantization groups along the input-channel dimension. At step k, let $\mathcal{G}_k$ be the current group, and let $\mathcal{R}_k := \bigcup_{t>k} \mathcal{G}_t$ be the set of later coordinates. Let $\mathbf{w}_k^{\text{eff}}$ be the current target of $\mathcal{G}_k$ after previous compensation steps, and let $\hat{\mathbf{w}}_k$ be its reconstructed version from the quantized value. The resulting quantization error is $\mathbf{e}_k := \mathbf{w}_k^{\text{eff}} - \hat{\mathbf{w}}_k$.

Consider the current Hessian for the quantization step k:

$$\mathbf{H}^{(k)} := \mathbf{H}_{\mathcal{G}_k\cup\mathcal{R}_k,\,\mathcal{G}_k\cup\mathcal{R}_k} = \begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{B}_k^\top & \mathbf{C}_k \end{bmatrix}, \tag{5}$$

where $\mathbf{A}_k{=}\mathbf{H}_{\mathcal{G}_k,\mathcal{G}_k}$ is the *within-group* block (blue 3×3 block in Fig. 2), $\mathbf{B}_k{=}\mathbf{H}_{\mathcal{G}_k,\mathcal{R}_k}$ is the *cross-group* block, and $\mathbf{C}_k{=}\mathbf{H}_{\mathcal{R}_k,\mathcal{R}_k}$ is the coordinates to be quantized later (Future Blocks in orange).

Let $\boldsymbol{\delta}_{\mathcal{R}_k} \in \mathbb{R}^{1\times|\mathcal{R}_k|}$ be real-valued compensation updates to the later coordinates before their quantization, and $\mathbf{X}_{\mathcal{G}_k}$ and $\mathbf{X}_{\mathcal{R}_k}$ be the columns of X corresponding to the group $\mathcal{G}_k$ and $\mathcal{R}_k$, respectively. For a fixed error $\mathbf{e}_k$, the local blockwise objective from the reconstruction loss in equation 3 becomes

$$\begin{aligned}\mathcal{L}_k(\boldsymbol{\delta}_{\mathcal{R}_k}) &= \|\mathbf{e}_k\mathbf{X}_{\mathcal{G}_k} - \boldsymbol{\delta}_{\mathcal{R}_k}\mathbf{X}_{\mathcal{R}_k}\|_2^2 = \mathbf{e}_k\mathbf{A}_k\mathbf{e}_k^\top - 2\mathbf{e}_k\mathbf{B}_k\boldsymbol{\delta}_{\mathcal{R}_k}^\top + \boldsymbol{\delta}_{\mathcal{R}_k}\mathbf{C}_k\boldsymbol{\delta}_{\mathcal{R}_k}^\top \\ &= \underbrace{\left(\boldsymbol{\delta}_{\mathcal{R}_k} - \mathbf{e}_k\mathbf{B}_k\mathbf{C}_k^{-1}\right)\mathbf{C}_k\left(\boldsymbol{\delta}_{\mathcal{R}_k} - \mathbf{e}_k\mathbf{B}_k\mathbf{C}_k^{-1}\right)^\top}_{\text{error compensation term}} + \underbrace{\mathbf{e}_k\left(\mathbf{A}_k - \mathbf{B}_k\mathbf{C}_k^{-1}\mathbf{B}_k^\top\right)\mathbf{e}_k^\top}_{\text{residual reconstruction loss}}.\end{aligned} \tag{6}$$

The loss has two terms: an *error-compensation term* and a *residual reconstruction loss*. Conventional methods (11; 12; 13) minimize the local objective by selecting the optimal $\boldsymbol{\delta}_{\mathcal{R}_k}$ for the first term while leaving the residual unchanged; instead, the residual serves solely to derive the theoretical error bound (13) for their method. The residual loss arises from components of the reconstruction error $\mathbf{e}X_{\mathcal{G}}$ that are orthogonal to the $X_{\mathcal{R}}$. These components cannot be reduced by adjusting the later weights $\mathbf{w}_{\mathcal{R}_k}$; it varies significantly depending on how well the current group is quantized, especially in low-bit settings where $\mathbf{e}_k$ is large. We directly account for this residual loss in Section 5.3.

We minimize the error-compensation term by setting $\boldsymbol{\delta}^\star_{\mathcal{R}_k} = \mathbf{e}_k\mathbf{B}_k\mathbf{C}_k^{-1}$. When the block size is one, this corresponds to the right-hand term in equation 4. In practice, the update $\boldsymbol{\delta}^\star_{\mathcal{R}_k}$ is computed from the inverse Hessian without explicitly computing $\mathbf{C}_k^{-1}$ at every step for the efficiency. The full derivation of equation 6 and its inverse-based form is provided in Appendix B.1.

The decomposition in equation 6 guides how we handle noisy off-diagonal Hessian estimates. Error compensation relies on cross-group curvature to propagate the quantization error from the current block; thus, Section 5.2 stabilizes this path by shrinking cross-group Hessian entries. However, the residual loss term is highly sensitive to the amount of shrinkage, as it depends quadratically on $\mathbf{e}_k$. We therefore use a conservative upper-bound surrogate, as explained in Section 5.3.

5.2 Optimal Cross-Group Shrinkage

We apply shrinkage to the off-diagonal entries of Hessian estimates to manage the uncertainty under limited calibration data, but do so in a group-wise manner. Specifically, the off-diagonal entries that directly affect forward error compensation are the blocks between quantization groups. We therefore apply shrinkage to the cross-group entries, as shown in Fig. 2, while preserving the within-group curvature for the residual-aware quantization in Section 5.3.

Let $\hat{\mathbf{H}} = \mathbf{X}\mathbf{X}^\top$ be the empirical Hessian, and let $\boldsymbol{\Sigma} = \mathbf{H}/N$ the true normalized covariance while $\hat{\boldsymbol{\Sigma}}$ is its estimate. Let $\gamma(p)$ denote the index of the quantization group containing coordinate p, and define $\mathcal{C} := \{(p,q) : \gamma(p) \neq \gamma(q)\}$, and let $\hat{\boldsymbol{\Sigma}}_C$ denote the restriction of $\hat{\boldsymbol{\Sigma}}$ to $\mathcal{C}$. Under the cross-group noise model $\hat{\boldsymbol{\Sigma}}_C = \boldsymbol{\Sigma}_C + \boldsymbol{\Xi}_C$, $\mathbb{E}[\boldsymbol{\Xi}_C] = \mathbf{0}$, $V_C := \mathbb{E}[\boldsymbol{\Xi}_C^2]$, we choose shrinkage coefficients by minimizing

$$\mathcal{L}(\rho) = \mathbb{E}\left[\left(\rho\,\hat{\Sigma}_{pq} - \Sigma_{pq}\right)^2\right] + \lambda_\rho\,\hat{\Sigma}_{pq}^2\,\rho^2. \tag{7}$$

Its closed-form solution is

$$\rho_{pq} = \frac{\boldsymbol{\Sigma}_{pq}^2}{\boldsymbol{\Sigma}_{pq}^2 + V_{pq} + \lambda_\rho\hat{\boldsymbol{\Sigma}}_{pq}^2}, \quad \rho^\star = \frac{1}{|\mathcal{C}|}\sum_{(p,q)\in\mathcal{C}}\rho_{pq}. \tag{8}$$

Because the resulting $\rho^\star$ averages the optimal shrinkages ρ_{pq} over all cross-group entries, it is less sensitive to individual off-diagonal fluctuations. In practice, let $\widehat{V}_{pq}$ denote the empirical variance estimate obtained from the calibration statistics, and define the noise-corrected signal estimate $\widehat{U}_{pq} := [\hat{\Sigma}^2_{pq} - \widehat{V}_{pq}]_+$, where $[x]_+ := \max(x, 0)$. We then compute

$$\widehat{\rho}_{pq} = \frac{\widehat{U}_{pq}}{\widehat{U}_{pq} + \widehat{V}_{pq} + \lambda_\rho \hat{\Sigma}^2_{pq}}, \qquad \widehat{\rho} = \frac{1}{|\mathcal{C}|} \sum_{(p,q)\in\mathcal{C}} \widehat{\rho}_{pq}, \qquad \tilde{\mathbf{H}} = \begin{cases} \hat{H}_{pq}, & \gamma(p) = \gamma(q), \\ \widehat{\rho}\, \hat{H}_{pq}, & \gamma(p) \neq \gamma(q). \end{cases} \tag{9}$$

Thus, uncertain cross-group entries are shrunk while the within-group curvature is preserved. Appendix B.2 provides the full derivation.

5.3 Residual-Aware Within-Group Quantization

We now consider within-group quantization using the weight target $\mathbf{w}_k^{\text{eff}}$ after previous compensation updates. As shown in equation 6, the residual reconstruction loss depends on how the current group is quantized. Directly minimizing this residual, however, relies on the estimated compensation benefit $\mathbf{B}_k\mathbf{C}_k^{-1}\mathbf{B}_k^\top$, which is affected by uncertain cross-group correlations. Using only the diagonal of $\mathbf{A}_k$ avoids these dependencies but ignores within-group cross terms and does not generally upper-bound the residual loss. We therefore preserve $\mathbf{A}_k$ and account for the magnitudes of its off-diagonal entries through a conservative, separable upper-bound objective:

$$d_j := (\mathbf{A}_k)_{jj} + \sum_{t\in\mathcal{G}_k, t\neq j} |(\mathbf{A}_k)_{jt}|, \qquad \mathbf{e}_k \left(\mathbf{A}_k - \mathbf{B}_k\mathbf{C}_k^{-1}\mathbf{B}_k^\top\right)\mathbf{e}_k^\top \leq \mathbf{e}_k\mathbf{A}_k\mathbf{e}_k^\top \leq \sum_{j\in\mathcal{G}_k} d_j e_{k,j}^2. \tag{10}$$

Here, $\mathbf{e}_k = \mathbf{w}_k^{\text{eff}} - \hat{\mathbf{w}}_k$. The additional row-sum term accounts for within-group cross terms using the absolute magnitudes of the off-diagonal entries. We use d_j to weight each coordinate in the within-group quantization objective, yielding a conservative upper bound on the residual reconstruction loss via the right inequality in Eq. 10. The derivation is provided in Appendix B.3. Unlike the cross-group entries used for compensation, these weights are constructed from the preserved within-group block and are unchanged by cross-group shrinkage; the formulation is examined in Appendix D.4.

The derived weights d_j define a common residual-aware objective for different quantization representations. Let $\mathcal{Q}_k$ denote the set of weight reconstructions allowed by the chosen quantizer. We formulate within-group quantization as

$$\min_{\hat{\mathbf{w}}_k} \sum_{j\in\mathcal{G}_k} d_j \left(w_{k,j}^{\text{eff}} - \hat{w}_{k,j}\right)^2, \qquad \text{subject to} \quad \hat{\mathbf{w}}_k \in \mathcal{Q}_k. \tag{11}$$

This formulation is general and applicable to uniform or non-uniform quantization methods. For example, we can apply this objective to the E8P lattice search of QuIP# (13) and the Viterbi trellis search of QTIP (14), using d_j to weight the coordinate-wise squared errors in their search costs. In both implementations, this weighted quantization is combined with the cross-group shrinkage described in Section 5.2. Appendix D.9 reports the corresponding experiments.

For our default implementation, we apply a practical group-wise affine quantization procedure previously used in DASH-Q (10), while replacing its feature weights with the upper-bound weights (d_j) to stably minimize the residual loss. With $\hat{w}_{k,j} = s \cdot q_j - z$, the objective becomes

$$\min_{s,z,Q} \sum_{j\in\mathcal{G}_k} d_j \left(w_{k,j}^{\text{eff}} - (s \cdot q_j - z)\right)^2 + \lambda_s s^2, \tag{12}$$

where $Q = \{q_j\}_{j\in\mathcal{G}_k}$ contains the quantized integers and $\lambda_s > 0$ is a small ridge coefficient. The closed-form solution for s and z under d_j and w^{eff} are derived in Appendix B.4, and the complete procedure is summarized in Appendix C. This implementation retains the group-wise affine representation while using the residual upper bound to determine the fitting weights.

6 Evaluation

6.1 Experimental Setup

We implement ReSC-Q using Python 3.12.3 and PyTorch 2.9.1, executing all experiments on a compute node equipped with an AMD EPYC 9755 CPU and an NVIDIA RTX PRO 6000 GPU under

Table 1: Performance comparison of ReSC-Q against seven affine PTQ baselines on three LLMs. We report WikiText-2 perplexity (PPL), zero-shot task accuracies, and total quantization time.

Method	W-bit	PPL ↓	Zero-Shot Reasoning Tasks									Time (s)
			ARC-C	ARC-E	BoolQ	Hella	OBQA	PIQA	SIQA	Wino	Avg ↑	
Llama-3.1-8B-Instruct												
Baseline	BF16	7.22	55.20	79.71	85.41	79.54	45.00	81.07	42.27	73.88	67.76	–
AWQ	4	7.53	54.01	78.58	84.31	**78.92**	43.60	80.14	42.27	**74.74**	**67.07**	4466.09
GPTQ	4	7.48	52.56	76.85	84.53	77.00	41.00	79.65	39.71	73.72	65.63	405.38
GPTQ + QEP	4	7.44	53.58	**79.21**	**85.63**	78.09	42.80	80.63	40.43	73.88	66.78	619.00
OmniQuant	4	7.66	**54.61**	78.75	83.98	78.38	44.00	80.20	**42.48**	73.16	66.94	4112.32
QuIP	4	7.46	52.30	77.90	85.17	78.79	43.60	80.41	41.20	74.35	66.72	705.70
OWQ	4.30	7.42	53.07	79.00	84.07	78.91	**44.20**	80.52	41.56	74.51	66.98	397.54
DASH-Q	4	7.55	53.75	78.32	83.91	78.57	43.80	**80.96**	42.43	73.40	66.89	84.87
ReSC-Q	4	**7.40**	53.92	78.79	84.86	78.58	43.40	**80.96**	41.40	73.09	66.88	502.90
AWQ	3	8.90	46.16	71.59	80.92	75.27	41.00	79.43	39.15	71.67	63.15	4458.15
GPTQ	3	8.33	24.15	44.28	39.82	37.55	28.60	60.94	34.14	53.99	40.43	405.99
GPTQ + QEP	3	8.18	46.59	71.72	81.47	72.40	37.20	76.06	35.16	69.38	61.25	623.00
OmniQuant	3	8.96	48.89	73.48	83.61	75.56	**43.20**	78.35	38.54	71.11	64.09	4145.30
QuIP	3	8.24	49.06	76.18	**84.01**	76.23	41.00	78.29	40.17	72.77	64.71	705.04
OWQ	3.32	8.10	48.63	74.75	82.57	**76.99**	42.00	78.89	39.51	72.61	64.49	397.64
DASH-Q	3	8.52	50.68	74.87	82.69	76.27	42.20	**80.14**	39.92	72.22	64.87	84.86
ReSC-Q	3	**8.05**	**53.50**	**79.55**	83.33	76.30	**43.20**	79.54	**40.84**	**73.80**	**66.26**	531.21
AWQ	2	128.34	26.11	33.21	50.12	36.38	24.00	57.51	34.08	50.04	38.93	4430.02
GPTQ	2	28.33	25.94	26.47	38.17	25.96	30.60	53.97	34.49	49.64	35.66	406.31
GPTQ + QEP	2	15.77	26.02	25.13	43.18	26.31	28.40	51.80	34.60	49.49	35.61	721.34
OmniQuant	2	54.22	25.34	39.14	45.96	43.39	29.20	61.97	34.29	54.54	41.73	4179.17
QuIP	2	21.20	26.02	36.07	50.06	37.01	28.00	57.07	33.32	51.62	39.90	700.24
OWQ	2.33	15.80	25.94	38.72	51.96	46.79	30.80	58.65	34.54	52.64	42.51	399.97
DASH-Q	2	17.17	39.25	67.05	76.15	62.09	34.60	73.45	33.98	66.93	56.69	69.71
ReSC-Q	2	**11.76**	**41.38**	**71.42**	**80.24**	**64.30**	**36.80**	**73.99**	**34.90**	**67.56**	**58.83**	588.09
Qwen3-14B												
Baseline	BF16	8.65	60.41	82.79	89.33	78.84	46.20	79.98	44.27	72.61	69.30	–
AWQ	2	15.12	37.54	61.07	68.29	61.67	35.40	71.60	37.10	57.22	53.74	7719.32
GPTQ	2	13.65	33.70	52.10	72.45	59.26	36.20	68.66	35.36	56.35	51.76	801.29
GPTQ + QEP	2	13.88	40.02	60.44	78.96	61.43	35.40	69.15	36.34	59.83	55.20	1219.08
OmniQuant	2	14.44	46.50	73.74	78.20	64.50	41.20	74.92	37.87	64.64	60.20	7479.48
QuIP	2	11.81	36.01	54.34	71.47	64.86	35.60	70.51	37.41	60.30	53.81	1553.46
OWQ	2.26	11.85	36.01	56.90	75.02	63.15	36.00	72.74	34.90	59.27	54.25	804.41
DASH-Q	2	11.41	53.33	**80.18**	84.59	**68.41**	41.60	75.57	39.25	69.30	64.03	119.21
ReSC-Q	2	**10.79**	**55.12**	79.08	**87.43**	67.78	**42.60**	**76.17**	**40.58**	**70.56**	**64.92**	1,235.20
Mixtral-8x7B-Instruct-v0.1												
Baseline	BF16	4.14	65.53	84.26	88.69	86.46	50.00	84.22	46.93	77.51	72.95	–
AWQ	2	7.53	43.60	66.25	69.91	65.21	36.00	76.71	36.03	57.54	56.41	9683.59
GPTQ	2	12.74	25.43	30.93	44.19	30.35	25.40	58.32	35.52	50.12	37.53	2313.41
GPTQ + QEP	2	9.41	28.41	36.91	51.01	32.02	25.80	58.16	34.54	51.70	39.82	4408.95
OmniQuant	2	6.30	47.61	72.14	76.02	63.19	33.20	78.89	36.54	56.99	58.07	20211.77
QuIP	2	5.94	53.24	74.75	80.09	78.47	43.80	79.43	41.86	73.24	65.61	4571.50
OWQ	2.31	5.80	51.28	74.62	77.68	72.65	40.00	79.00	36.90	61.40	61.69	2288.56
DASH-Q	2	5.51	**58.79**	**81.19**	87.22	**80.33**	46.20	81.88	45.19	77.35	69.77	235.46
ReSC-Q	2	**5.22**	58.02	81.02	**87.83**	79.97	**47.00**	**82.48**	**45.70**	**77.43**	**69.93**	4290.54

Table 2: Performance on large models (≥70B) under 2-bit quantization (same notation as Table 1).

Method	Llama-3.1-70B-Instruct			GLM-4.5-Air			Qwen3.5-122B-A10B			Mistral-Large-Instruct-2407		
	PPL ↓	Acc. ↑	Size (GB)	PPL ↓	Acc. ↑	Size (GB)	PPL ↓	Acc. ↑	Size (GB)	PPL ↓	Acc. ↑	Size (GB)
BF16	3.78	73.23	131.42	4.87	71.93	199.03	5.07	71.05	227.45	2.74	74.27	228.38
GPTQ	168.49	35.20	27.82	94.48	45.18	39.23	–	33.78	45.02	4.94	63.37	44.04
GPTQ+QEP	-	34.56	27.82	52.94	54.29	39.23	8.01	65.30	45.02	5.20	64.01	44.04
OWQ	9.94	59.31	29.17	7.55	65.75	47.42	7.14	65.46	57.90	4.80	69.03	45.75
DASH-Q	7.97	67.47	27.82	7.46	69.99	39.23	8.05	68.62	45.02	4.43	70.86	44.04
ReSC-Q	**7.28**	**68.79**	27.82	**7.43**	**70.38**	39.23	**6.87**	**70.06**	45.02	**4.14**	**72.18**	44.04

CUDA 13.0. We evaluate 13 LLMs: MobileLLM family(125M, 350M, 600M, 1B, 1.5B) (25), Llama-3.1-8B-Instruct, Llama-3.1-70B-Instruct (26), Qwen3-14B (27), Mixtral-8x7B-Instruct-v0.1 (28), GLM-4.5-Air (29), Qwen3.5-122B-A10B (30), and Mistral-Large-Instruct-2407 (31) for accuracy, and Mistral-7B-Instruct-v0.3 (32) for qualitative analysis in Appendix. For each model and methods, we perform calibration using 128 samples with each sample with sequence length of 2048 randomly drawn from the WikiText-2 (33) training set and evaluate perplexity on the test set. General reasoning capabilities are measured across eight zero-shot tasks—ARC (Easy/Challenge) (34), PIQA (35),

Hellaswag (36), Winogrande (37), BoolQ (38), SIQA (39), and OpenBookQA (40) —via the LM Evaluation Harness (41). All experiments perform weight-only quantization with group sizes of 128 for 4-bit, 64 for 3-bit, and 32 for 2-bit precision.

We implement all baselines using their official repositories and method-specific configurations. We use each baseline's recommended configuration to ensure a fair comparison. ReSC-Q runs with $\lambda_\rho = 10^{-2}$; for DASH-Q we set $T = 9$ and $\lambda_s = 10^{-2}$. AWQ uses a grid-search size of 20; GPTQ+QEP uses the recommended `perccorr` coefficient $\gamma = 0.5$; and OmniQuant enables *learnable weight clipping* (LWC) with 20 optimization epochs and a learning rate of 10^{-2}. GPTQ, QuIP, and OWQ use a block size of 128, with 128 retained outliers for OWQ. Affine scales and zero-points are stored in FP16 for all methods.

6.2 Overall Accuracy

Table 1 compares ReSC-Q with seven affine PTQ baselines on three LLMs, reporting WikiText-2 perplexity, zero-shot accuracy, and quantization time. On `Llama-3.1-8B`, ReSC-Q remains competitive at 4-bit and achieves the lowest perplexity and highest average zero-shot accuracy at 3-bit and 2-bit. The same advantage is observed in the 2-bit results on `Qwen3-14B` and `Mixtral-8x7B`, where ReSC-Q outperforms the compared baselines in both metrics.

Table 2 extends the evaluation to four larger models ($\geq$70B), reporting perplexity, average zero-shot accuracy, and model size. ReSC-Q achieves the lowest perplexity and highest average zero-shot accuracy among the compared quantized methods on all four models. These results show that its low-bit performance advantage extends to larger model scales while retaining the standard group-wise affine representation. The full task-wise results are provided in Appendix D.1. Overall, across the seven models in Tables 1 and 2, ReSC-Q improves 2-bit average zero-shot accuracy by 1.09 pp over the strongest baseline, reducing the mean accuracy gap to full precision by 23.2%.

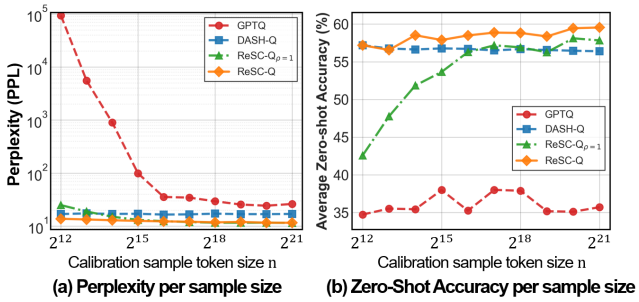

Figure 3: Sensitivity to calibration size on `Llama-3.1-8B-Instruct` at 2-bit. (a) WikiText-2 perplexity and (b) average zero-shot accuracy as the number of calibration tokens increases. ReSC-Q$_{\rho=1}$ fixes the cross-group shrinkage coefficient to one.

6.3 Analysis on Sensitivity to Calibration Dataset Size

We evaluate calibration-size sensitivity on `Llama-3.1-8B-Instruct`. Fig. 3(a) and (b) compare GPTQ, DASH-Q, ReSC-Q$_{\rho=1}$, and ReSC-Q. The $\rho = 1$ variant retains the same residual-aware within-group objective and blockwise error compensation as ReSC-Q, but omits cross-group shrinkage. Detailed results are provided in Appendix D.3.

GPTQ is highly sensitive to calibration size, with severe perplexity degradation at small budgets and consistently low zero-shot accuracy. DASH-Q remains relatively stable across calibration sizes, but without error compensation, it consistently underperforms ReSC-Q in both perplexity (up to 5.4 higher) and zero-shot accuracy (up to 3.2 pp lower). ReSC-Q$_{\rho=1}$ improves as more calibration data become available, but its zero-shot accuracy drops substantially at small budgets despite retaining the residual-aware quantization objective. This behavior suggests that controlling the within-group residual alone is insufficient when compensation relies on uncertain cross-group estimates. ReSC-Q reduces this low-budget degradation and maintains high zero-shot accuracy across calibration sizes. At larger budgets, its perplexity becomes comparable to that of ReSC-Q$_{\rho=1}$, while its average zero-

Table 3: Comparison of calibration-estimated layer-wise shrinkage against oracle-selected global coefficients on Llama-3.1-8B-Instruct. Each entry reports PPL↓ / average zero-shot accuracy↑. The last column reports the mean and layer-wise range of the estimated coefficient $\hat{\rho}$.

n	$\rho = 0.0$	$\rho = 0.2$	$\rho = 0.4$	$\rho = 0.6$	$\rho = 0.8$	$\rho = 1.0$	**ReSC-Q**	**Mean $\hat{\rho}$ [min, max]**
2^{12}	17.15 / 56.3	**13.85** / 57.0	13.94 / **57.5**	14.18 / 56.3	14.90 / 55.6	24.87 / 42.57	13.73 / 57.20	0.46 [0.40, 0.54]
2^{13}	17.00 / 55.1	13.47 / **57.3**	**13.23** / 55.8	13.40 / 56.3	13.74 / 56.5	18.84 / 47.75	13.34 / 56.57	0.43 [0.35, 0.53]
2^{14}	17.48 / 56.0	13.79 / 57.7	**12.95** / **58.8**	13.06 / 57.0	13.08 / 56.4	15.02 / 51.85	12.89 / 58.52	0.42 [0.33, 0.54]
2^{15}	17.18 / 55.8	13.14 / 58.0	**12.72** / 58.2	12.85 / **58.7**	12.90 / 57.6	13.10 / 53.65	12.54 / 57.89	0.49 [0.39, 0.62]
2^{16}	17.13 / 55.0	13.21 / 58.0	12.65 / **58.9**	**12.30** / 58.4	12.32 / 58.5	12.38 / 56.29	12.28 / 58.50	0.59 [0.48, 0.71]
2^{17}	17.56 / 55.4	13.60 / **58.5**	12.70 / 58.1	12.06 / **58.5**	**11.92** / 57.8	11.93 / 57.18	12.09 / 58.87	0.68 [0.58, 0.78]
2^{18}	17.44 / 56.1	13.45 / 58.3	12.41 / **58.7**	12.20 / 58.2	11.82 / 58.3	**11.58** / 56.91	11.76 / 58.83	0.74 [0.64, 0.83]
2^{19}	17.24 / 55.9	13.47 / 57.5	12.44 / **59.1**	11.93 / **59.1**	11.85 / **59.1**	**11.57** / 56.27	11.96 / 58.36	0.81 [0.72, 0.88]

shot accuracy remains higher. Together, these results support combining residual-aware quantization with cross-group shrinkage to reduce calibration sensitivity while retaining useful error compensation.

6.4 Generalization of Calibration-based Layer-wise Shrinkage

Table 3 evaluates whether ReSC-Q's calibration-based, layer-wise shrinkage generalizes to unseen data, reporting it alongside oracle-selected global coefficients based on unseen-data performance on `Llama-3.1-8B-Instruct` at 2-bit. Global coefficients are shared across all layers, whereas ReSC-Q estimates one coefficient per layer from calibration statistics. We report perplexity, average zero-shot accuracy, and the mean and range of the estimated coefficients.

With limited calibration data, $\hat{\rho}$ takes intermediate values, reducing reliance on uncertain cross-group estimates. As calibration size grows, $\hat{\rho}$ generally increases, retaining more cross-group information for compensation. Global-coefficient results show a similar trend in perplexity: larger coefficients become more effective with more calibration data. However, the coefficient that minimizes perplexity does not always maximize zero-shot accuracy; the best global setting depends on both calibration size and evaluation metric. ReSC-Q performs close to the best global choices on both metrics.

7 Limitations

The shrinkage coefficient is derived under a simplified cross-group noise model with zero-mean estimation error. This model captures sampling variance but not potential underlying bias. In practice, calibration samples can differ substantially from the deployment distribution, and the resulting uncertainty may not be fully captured by this model. Consequently, the estimated shrinkage coefficient may under- or overestimate the reliability of cross-group curvature. Also like other reconstruction-based PTQ methods, ReSC-Q optimizes a layer reconstruction objective rather than downstream task loss. Although our experiments show strong downstream accuracy, improvements in local reconstruction do not necessarily translate monotonically to generation quality or task accuracy.

8 Conclusion

We presented ReSC-Q, a Hessian-based PTQ method that addresses the reliability of error compensation under limited calibration data. Our analysis shows that off-diagonal Hessian estimates used for compensation are less reliable than diagonal entries, and that compensation optimized from calibration data does not always reduce reconstruction loss on unseen inputs. ReSC-Q addresses this problem by applying shrinkage to uncertain cross-group curvature, while explicitly accounting for the reconstruction loss that remains after compensation. For within-group quantization, ReSC-Q derives a separable upper-bound objective for the post-compensation residual, which can be incorporated into different quantization representations; we use group-wise affine quantization as the default implementation. Across seven LLMs evaluated at 2-bit, ReSC-Q improves average zero-shot accuracy by 1.09 percentage points over the strongest baseline for each model, reducing the mean accuracy gap to full precision by 23.2%. These results show that controlling uncertain error compensation while accounting for its remaining residual provides a reliable approach to ultra-low-bit PTQ.

REFERENCES

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

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

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

[4] Changhun Lee, Jungyu Jin, Taesu Kim, Hyungjun Kim, and Eunhyeok Park. Owq: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13355–13364, 2024.

[5] Zhihang Yuan, Lin Niu, Jiawei Liu, Wenyu Liu, Xinggang Wang, Yuzhang Shang, Guangyu Sun, Qiang Wu, Jiaxiang Wu, and Bingzhe Wu. Rptq: Reorder-based post-training quantization for large language models. *arXiv preprint arXiv:2304.01089*, 2023.

[6] Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, and Baris Kasikci. Atom: Low-bit quantization for efficient and accurate llm serving. *Proceedings of Machine Learning and Systems*, 6:196–209, 2024.

[7] Jaemin Kim, Hongjun Um, Sungkyun Kim, Yongjun Park, and Jiwon Seo. Flexiq: Adaptive mixed-precision quantization for latency/accuracy trade-offs in deep neural networks. *arXiv preprint arXiv:2510.02822*, 2025.

[8] Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W Mahoney, and Kurt Keutzer. Squeezellm: Dense-and-sparse quantization. *arXiv preprint arXiv:2306.07629*, 2023.

[9] Tianyi Zhang and Anshumali Shrivastava. Leanquant: Accurate and scalable large language model quantization with loss-error-aware grid. In *International Conference on Learning Representations*, volume 2025, pages 35521–35544, 2025.

[10] Jaemin Kim, Sungkyun Kim, Junyeol Lee, and Jiwon Seo. Robust ultra low-bit post-training quantization via stable diagonal curvature estimate. In *Proceedings of the Sixth European Workshop on Machine Learning and Systems*, pages 274–285, 2026.

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

[12] Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. Quip: 2-bit quantization of large language models with guarantees. *Advances in Neural Information Processing Systems*, 36:4396–4429, 2023.

[13] Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks. *Proceedings of machine learning research*, 235:48630, 2024.

[14] Albert Tseng, Qingyao Sun, David Hou, and Christopher De. Qtip: Quantization with trellises and incoherence processing. *Advances in Neural Information Processing Systems*, 37:59597–59620, 2024.

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

[16] Vage Egiazarian, Roberto L Castro, Denis Kuznedelev, Andrei Panferov, Eldar Kurtic, Shubhra Pandit, Alexandre Marques, Mark Kurtz, Saleh Ashkboos, Torsten Hoefler, et al. Bridging the gap between promise and performance for microscaling fp4 quantization. *arXiv preprint arXiv:2509.23202*, 2025.

[17] Jinhao Zhang Yunquan Zhang, Boyang Zhang, Jun Sun, Daning Cheng, et al. Hero-q: A general framework for stable low bit quantization via hessian conditioning. *arXiv e-prints*, pages arXiv–2601, 2026.

[18] Jemin Lee, Sihyeong Park, Jinse Kwon, Jihun Oh, and Yongin Kwon. Exploring the trade-offs: Quantization methods, task difficulty, and model size in large language models from edge to giant. *arXiv preprint arXiv:2409.11055*, 2024.

[19] Yamato Arai and Yuma Ichikawa. Quantization error propagation: Revisiting layer-wise post-training quantization. *arXiv preprint arXiv:2504.09629*, 2025.

[20] Elias Frantar and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning. *Advances in Neural Information Processing Systems*, 35:4475–4488, 2022.

[21] Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411, 2004.

[22] Richard H Byrd, Samantha L Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.

[23] Alexander Soen and Ke Sun. On the variance of the fisher information for deep learning. *Advances in Neural Information Processing Systems*, 34:5708–5719, 2021.

[24] Alexander Soen and Ke Sun. Trade-offs of diagonal fisher information matrix estimators. *Advances in Neural Information Processing Systems*, 37:5870–5912, 2024.

[25] Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, et al. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases. In *Forty-first International Conference on Machine Learning*, 2024.

[26] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

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

[28] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[29] Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025.

[30] Qwen Team. Qwen3. 5-omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

[31] Mistral AI. Large enough. `https://mistral.ai/news/mistral-large-2407`, 2024. Official Mistral Large 2 release blog. Accessed: 2026-05-04.

[32] Albert Q Jiang, A Sablayrolles, A Mensch, C Bamford, D Singh Chaplot, Ddl Casas, F Bressand, G Lengyel, G Lample, L Saulnier, et al. Mistral 7b. arxiv. *arXiv preprint arXiv:2310.06825*, 10(3), 2023.

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

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

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

[36] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

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

[38] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

[39] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

[40] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

[41] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024.

A Disclosure of AI Assistance

We used generative AI tools solely for language polishing and figure design, with assistance limited to improving the clarity and readability of the manuscript and the visual presentation of figures. Generative AI tools were not used for research ideation, method development, theoretical analysis, implementation of the proposed methods, experimental design or execution, data generation, or analysis and interpretation of results. The authors reviewed and verified all AI-assisted text and figures and take full responsibility for the final content of this paper.

B Additional Derivations

Consider a linear layer with weight $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ and calibration inputs $\mathbf{X} \in \mathbb{R}^{d_{\text{in}} \times N_{\text{token}}}$, where each column of $\mathbf{X}$ is an input activation token. Weight-only PTQ seeks a quantized reconstruction $\widehat{\mathbf{W}}$ that preserves the layer output on the calibration data:

$$\mathcal{L}(\widehat{\mathbf{W}}) = \left\| \mathbf{W}\mathbf{X} - \widehat{\mathbf{W}}\mathbf{X} \right\|_F^2. \tag{13}$$

Writing $\mathbf{w}_i = \mathbf{W}_{i,:}$ and $\widehat{\mathbf{w}}_i = \widehat{\mathbf{W}}_{i,:}$ for the i-th output row, equation 13 decomposes as

$$\begin{aligned} \mathcal{L}(\widehat{\mathbf{W}}) &= \sum_{i=1}^{d_{\text{out}}} \|(\mathbf{w}_i - \widehat{\mathbf{w}}_i)\mathbf{X}\|_2^2 \\ &= \sum_{i=1}^{d_{\text{out}}} (\mathbf{w}_i - \widehat{\mathbf{w}}_i)\widehat{\mathbf{H}}(\mathbf{w}_i - \widehat{\mathbf{w}}_i)^\top, \qquad \widehat{\mathbf{H}} := \mathbf{X}\mathbf{X}^\top. \end{aligned} \tag{14}$$

Thus the layer-wise objective is separable across output channels. All derivations below can therefore be written for a single output row and then applied independently to every row of the weight matrix.

Following equation 14, we derive the following expressions for a single output row and omit the output-row index unless stated otherwise. All weight, reconstruction, and error vectors are written as row vectors.

Coordinates refer to input-channel indices, i.e., columns of the weight matrix. For an index set $\mathcal{I}$, $\mathbf{v}_{\mathcal{I}}$ denotes the restriction of a row vector $\mathbf{v}$ to the coordinates in $\mathcal{I}$. For two index sets $\mathcal{I}$ and $\mathcal{J}$, $\mathbf{H}_{\mathcal{I},\mathcal{J}}$ denotes the corresponding block of a Hessian matrix.

Let $\{\mathcal{G}_k\}_{k=1}^m$ be the quantization groups along the input-channel dimension. At group step k, the groups before $\mathcal{G}_k$ have already been quantized and are held fixed. We define

$$\mathcal{R}_k := \bigcup_{t>k} \mathcal{G}_t, \qquad \mathcal{I}_k := \mathcal{G}_k \cup \mathcal{R}_k. \tag{15}$$

Here $\mathcal{R}_k$ is the set of later coordinates, and $\mathcal{I}_k$ is the *active* coordinate set, meaning the coordinates that have not yet been finalized at step k. Terms depending only on already finalized coordinates are constants for the current step and are omitted from the step-wise objectives below.

Unless otherwise specified, $\mathbf{H}$ denotes the unknown true Hessian for the corresponding derivation. For the propagation step of ReSC-Q, this is the shrunk propagation Hessian $\widetilde{\mathbf{H}}$. Since cross-group shrinkage leaves the within-group blocks unchanged, the within-group curvature used to construct the residual-aware quantization objective coincides with the corresponding block of the empirical Hessian.

B.1 Derivation of Blockwise Error Compensation

This appendix derives the blockwise forward compensation step used in Section 5.1. The derivation is row-wise; in implementation, the same process is applied to all output rows.

At step k, let $\mathbf{w}_k^{\text{eff}}$ be the current target values on the group $\mathcal{G}_k$ after all previous compensation updates, and let $\hat{\mathbf{w}}_k$ be their quantized reconstruction. The quantization error introduced by the current group is

$$\mathbf{e}_k = \mathbf{w}_k^{\text{eff}} - \hat{\mathbf{w}}_k \in \mathbb{R}^{1\times|\mathcal{G}_k|}. \tag{16}$$

Before the later groups are quantized, ReSC-Q allows a real-valued update

$$\boldsymbol{\delta}_{\mathcal{R}_k} \in \mathbb{R}^{1\times|\mathcal{R}_k|}$$

to their temporary targets. That is, the target on the later coordinates is changed from $\mathbf{w}_{\mathcal{R}_k}^{\text{eff}}$ to $\mathbf{w}_{\mathcal{R}_k}^{\text{eff}} + \boldsymbol{\delta}_{\mathcal{R}_k}$. Relative to the pre-update target, this contributes

$$\mathbf{w}_{\mathcal{R}_k}^{\text{eff}} - \left(\mathbf{w}_{\mathcal{R}_k}^{\text{eff}} + \boldsymbol{\delta}_{\mathcal{R}_k}\right) = -\boldsymbol{\delta}_{\mathcal{R}_k} \tag{17}$$

to the current reconstruction error. Therefore, on the active coordinate set ordered as $\mathcal{I}_k = [\mathcal{G}_k, \mathcal{R}_k]$, the step-wise error vector is

$$\mathbf{e}^{(k)} = [\mathbf{e}_k,\ -\boldsymbol{\delta}_{\mathcal{R}_k}]. \tag{18}$$

Since $\mathbf{e}^{(k)}$ has support only on $\mathcal{I}_k$, the part of the quadratic form that depends on the current decision only involves the principal Hessian block restricted to $\mathcal{I}_k$:

$$\mathbf{H}^{(k)} := \mathbf{H}_{\mathcal{I}_k,\mathcal{I}_k} = \begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{B}_k^\top & \mathbf{C}_k \end{bmatrix}, \tag{19}$$

where

$$\mathbf{A}_k = \mathbf{H}_{\mathcal{G}_k,\mathcal{G}_k}, \qquad \mathbf{B}_k = \mathbf{H}_{\mathcal{G}_k,\mathcal{R}_k}, \qquad \mathbf{C}_k = \mathbf{H}_{\mathcal{R}_k,\mathcal{R}_k}. \tag{20}$$

Here $\mathbf{A}_k$ is the within-current-group curvature, $\mathbf{B}_k$ couples the current group to the later coordinates, and $\mathbf{C}_k$ is the curvature among the later coordinates.

Substituting equation 18 into the row-wise quadratic approximation gives the step-wise objective

$$\begin{aligned} \mathcal{L}_k(\boldsymbol{\delta}_{\mathcal{R}_k}) &= \mathbf{e}^{(k)}\mathbf{H}^{(k)}\mathbf{e}^{(k)\top} \\ &= [\mathbf{e}_k,\ -\boldsymbol{\delta}_{\mathcal{R}_k}] \begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{B}_k^\top & \mathbf{C}_k \end{bmatrix} \begin{bmatrix} \mathbf{e}_k^\top \\ -\boldsymbol{\delta}_{\mathcal{R}_k}^\top \end{bmatrix} \\ &= \mathbf{e}_k\mathbf{A}_k\mathbf{e}_k^\top - 2\mathbf{e}_k\mathbf{B}_k\boldsymbol{\delta}_{\mathcal{R}_k}^\top + \boldsymbol{\delta}_{\mathcal{R}_k}\mathbf{C}_k\boldsymbol{\delta}_{\mathcal{R}_k}^\top. \end{aligned} \tag{21}$$

The propagation Hessian $\mathbf{H}^{(k)}$ is positive definite in practice; hence every principal block, including $\mathbf{C}_k$, is positive definite. Starting from equation 21, we add and subtract $\mathbf{e}\mathbf{B}_k\mathbf{C}_k^{-1}\mathbf{B}_k^\top\mathbf{e}^\top$:

$$\begin{aligned} \mathcal{L}_k(\boldsymbol{\delta}_{\mathcal{R}_k}) &= \mathbf{e}_k\mathbf{A}_k\mathbf{e}_k^\top - 2\mathbf{e}_k\mathbf{B}_k\boldsymbol{\delta}_{\mathcal{R}_k}^\top + \boldsymbol{\delta}_{\mathcal{R}_k}\mathbf{C}_k\boldsymbol{\delta}_{\mathcal{R}_k}^\top \\ &= \mathbf{e}_k\left(\mathbf{A}_k - \mathbf{B}_k\mathbf{C}_k^{-1}\mathbf{B}_k^\top\right)\mathbf{e}_k^\top \\ &\quad + \left[\boldsymbol{\delta}_{\mathcal{R}_k}\mathbf{C}_k\boldsymbol{\delta}_{\mathcal{R}_k}^\top - 2\mathbf{e}_k\mathbf{B}_k\boldsymbol{\delta}_{\mathcal{R}_k}^\top + \mathbf{e}_k\mathbf{B}_k\mathbf{C}_k^{-1}\mathbf{B}_k^\top\mathbf{e}_k^\top\right]. \end{aligned} \tag{22}$$

The bracketed term can be written as a quadratic square. Indeed, since $\mathbf{C}_k$ is symmetric,

$$\begin{aligned} &\boldsymbol{\delta}_{\mathcal{R}_k}\mathbf{C}_k\boldsymbol{\delta}_{\mathcal{R}_k}^\top - 2\mathbf{e}_k\mathbf{B}_k\boldsymbol{\delta}_{\mathcal{R}_k}^\top + \mathbf{e}_k\mathbf{B}_k\mathbf{C}_k^{-1}\mathbf{B}_k^\top\mathbf{e}_k^\top \\ &= \boldsymbol{\delta}_{\mathcal{R}_k}\mathbf{C}_k\boldsymbol{\delta}_{\mathcal{R}_k}^\top - \mathbf{e}_k\mathbf{B}_k\boldsymbol{\delta}_{\mathcal{R}_k}^\top - \boldsymbol{\delta}_{\mathcal{R}_k}\mathbf{B}_k^\top\mathbf{e}_k^\top + \mathbf{e}_k\mathbf{B}_k\mathbf{C}_k^{-1}\mathbf{B}_k^\top\mathbf{e}_k^\top \\ &= \left(\boldsymbol{\delta}_{\mathcal{R}_k} - \mathbf{e}_k\mathbf{B}_k\mathbf{C}_k^{-1}\right)\mathbf{C}_k\left(\boldsymbol{\delta}_{\mathcal{R}_k} - \mathbf{e}_k\mathbf{B}_k\mathbf{C}_k^{-1}\right)^\top. \end{aligned} \tag{23}$$

Combining equation 22 and equation 23, we obtain

$$\begin{aligned} \mathcal{L}_k(\boldsymbol{\delta}_{\mathcal{R}_k}) &= \left(\boldsymbol{\delta}_{\mathcal{R}_k} - \mathbf{e}_k\mathbf{B}_k\mathbf{C}_k^{-1}\right)\mathbf{C}_k\left(\boldsymbol{\delta}_{\mathcal{R}_k} - \mathbf{e}_k\mathbf{B}_k\mathbf{C}_k^{-1}\right)^\top \\ &\quad + \mathbf{e}_k\left(\mathbf{A}_k - \mathbf{B}_k\mathbf{C}_k^{-1}\mathbf{B}_k^\top\right)\mathbf{e}_k^\top. \end{aligned} \tag{24}$$

Since $\mathbf{C}_k \succ \mathbf{0}$, the first term in equation 24 is minimized when it is zero, giving the optimal compensation update

$$\boldsymbol{\delta}^{\star}_{\mathcal{R}_k} = \mathbf{e}_k \mathbf{B}_k \mathbf{C}_k^{-1}. \tag{25}$$

Although this update makes the error-compensation term zero, the local reconstruction loss still contains the second term, $\mathbf{e}_k(\mathbf{A}_k - \mathbf{B}_k \mathbf{C}_k^{-1} \mathbf{B}_k^{\top})\mathbf{e}_k^{\top}$. We refer to this remaining term as the *residual reconstruction loss*.

To explain this residual, consider the empirical reconstruction Hessian $\widehat{\mathbf{H}} = \mathbf{X}\mathbf{X}^{\top}$. Let $\mathbf{X}_{\mathcal{G}_k}$ and $\mathbf{X}_{\mathcal{R}_k}$ denote the calibration activations corresponding to the current and later coordinates, respectively. For this interpretation, take $\mathbf{A}_k$, $\mathbf{B}_k$, and $\mathbf{C}_k$ from the corresponding blocks of $\widehat{\mathbf{H}}$, with $\mathbf{C}_k \succ \mathbf{0}$:

$$\mathbf{A}_k = \mathbf{X}_{\mathcal{G}_k} \mathbf{X}_{\mathcal{G}_k}^{\top}, \qquad \mathbf{B}_k = \mathbf{X}_{\mathcal{G}_k} \mathbf{X}_{\mathcal{R}_k}^{\top}, \qquad \mathbf{C}_k = \mathbf{X}_{\mathcal{R}_k} \mathbf{X}_{\mathcal{R}_k}^{\top}. \tag{26}$$

The current group introduces an output error $\mathbf{e}_k \mathbf{X}_{\mathcal{G}_k}$, while the compensation update changes the output by $\boldsymbol{\delta}_{\mathcal{R}_k} \mathbf{X}_{\mathcal{R}_k}$. Thus, equation 21 can be written as

$$\mathcal{L}_k(\boldsymbol{\delta}_{\mathcal{R}_k}) = \|\mathbf{e}_k \mathbf{X}_{\mathcal{G}_k} - \boldsymbol{\delta}_{\mathcal{R}_k} \mathbf{X}_{\mathcal{R}_k}\|_2^2. \tag{27}$$

Let $\mathbf{r}_k$ denote the output error remaining after the optimal compensation:

$$\mathbf{r}_k := \mathbf{e}_k \mathbf{X}_{\mathcal{G}_k} - \boldsymbol{\delta}^{\star}_{\mathcal{R}_k} \mathbf{X}_{\mathcal{R}_k}. \tag{28}$$

Substituting equation 25 gives

$$\begin{aligned} \mathbf{r}_k \mathbf{X}_{\mathcal{R}_k}^{\top} &= \mathbf{e}_k \mathbf{X}_{\mathcal{G}_k} \mathbf{X}_{\mathcal{R}_k}^{\top} - \boldsymbol{\delta}^{\star}_{\mathcal{R}_k} \mathbf{X}_{\mathcal{R}_k} \mathbf{X}_{\mathcal{R}_k}^{\top} \\ &= \mathbf{e}_k \mathbf{B}_k - \boldsymbol{\delta}^{\star}_{\mathcal{R}_k} \mathbf{C}_k \\ &= \mathbf{e}_k \mathbf{B}_k - \mathbf{e}_k \mathbf{B}_k \mathbf{C}_k^{-1} \mathbf{C}_k = \mathbf{0}. \end{aligned} \tag{29}$$

Therefore, $\mathbf{r}_k$ is orthogonal to the row span of $\mathbf{X}_{\mathcal{R}_k}$. Since $\boldsymbol{\delta}^{\star}_{\mathcal{R}_k} \mathbf{X}_{\mathcal{R}_k}$ lies in this row span, the current group's output reconstruction error is decomposed into two orthogonal components:

$$\underbrace{\mathbf{e}_k \mathbf{X}_{\mathcal{G}_k}}_{\substack{\text{error caused by} \\ \text{the current group}}} = \underbrace{\boldsymbol{\delta}^{\star}_{\mathcal{R}_k} \mathbf{X}_{\mathcal{R}_k}}_{\substack{\text{component explainable by} \\ \mathbf{X}_{\mathcal{R}_k}}} + \underbrace{\mathbf{r}_k}_{\substack{\text{component orthogonal to} \\ \text{the row span of } \mathbf{X}_{\mathcal{R}_k}}}. \tag{30}$$

Any further adjustment to the later weights changes the output only within the row span of $\mathbf{X}_{\mathcal{R}_k}$ and therefore cannot reduce $\mathbf{r}_k$. Indeed, for any $\boldsymbol{\delta}_{\mathcal{R}_k}$, orthogonality gives

$$\begin{aligned} \mathcal{L}_k(\boldsymbol{\delta}_{\mathcal{R}_k}) &= \left\|\mathbf{r}_k + \left(\boldsymbol{\delta}^{\star}_{\mathcal{R}_k} - \boldsymbol{\delta}_{\mathcal{R}_k}\right) \mathbf{X}_{\mathcal{R}_k}\right\|_2^2 \\ &= \|\mathbf{r}_k\|_2^2 + \left\|\left(\boldsymbol{\delta}^{\star}_{\mathcal{R}_k} - \boldsymbol{\delta}_{\mathcal{R}_k}\right) \mathbf{X}_{\mathcal{R}_k}\right\|_2^2 \geq \|\mathbf{r}_k\|_2^2. \end{aligned} \tag{31}$$

Consequently, the minimized current-group loss is

$$\begin{aligned} \min_{\boldsymbol{\delta}_{\mathcal{R}_k}} \mathcal{L}_k(\boldsymbol{\delta}_{\mathcal{R}_k}) &= \|\mathbf{r}_k\|_2^2 \\ &= \|\mathbf{e}_k \mathbf{X}_{\mathcal{G}_k}\|_2^2 - \left\|\boldsymbol{\delta}^{\star}_{\mathcal{R}_k} \mathbf{X}_{\mathcal{R}_k}\right\|_2^2 \\ &= \mathbf{e}_k \left(\mathbf{A}_k - \mathbf{B}_k \mathbf{C}_k^{-1} \mathbf{B}_k^{\top}\right) \mathbf{e}_k^{\top} = \mathbf{e}_k \mathbf{S}_k \mathbf{e}_k^{\top}, \end{aligned} \tag{32}$$

where $\mathbf{S}_k := \mathbf{A}_k - \mathbf{B}_k \mathbf{C}_k^{-1} \mathbf{B}_k^{\top}$ is the Schur complement of $\mathbf{C}_k$ in $\mathbf{H}^{(k)}$. It is the curvature seen by the current-group error after the later coordinates have been optimally adjusted by a real-valued forward update. The minimum in equation 32 is obtained for a fixed current-group error $\mathbf{e}_k$. Changing how the current group is quantized changes $\mathbf{e}_k$ and can therefore change the residual reconstruction loss, even after optimal compensation. Appendix B.3 derives a separable upper-bound objective for this residual when quantizing the current group.

We next rewrite equation 25 in the inverse-Hessian form used by the implementation. Explicitly forming $\mathbf{C}_k^{-1}$ at every group step would be inefficient. Instead, ReSC-Q maintains the inverse of the current active Hessian block:

$$\mathbf{M}^{(k)} := (\mathbf{H}^{(k)})^{-1} = \begin{bmatrix} \mathbf{E}_k & \mathbf{F}_k \\ \mathbf{F}_k^{\top} & \mathbf{G}_k \end{bmatrix}, \tag{33}$$

where the block partition follows the same ordering $\mathcal{I}_k = [\mathcal{G}_k, \mathcal{R}_k]$.

To relate these blocks to equation 25, compare blocks in $\mathbf{H}^{(k)}\mathbf{M}^{(k)} = \mathbf{I}$:

$$\begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{B}_k^\top & \mathbf{C}_k \end{bmatrix} \begin{bmatrix} \mathbf{E}_k & \mathbf{F}_k \\ \mathbf{F}_k^\top & \mathbf{G}_k \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{34}$$

The lower-left block gives

$$\mathbf{B}_k^\top \mathbf{E}_k + \mathbf{C}_k \mathbf{F}_k^\top = \mathbf{0}, \qquad \text{so} \qquad \mathbf{F}_k^\top = -\mathbf{C}_k^{-1}\mathbf{B}_k^\top \mathbf{E}_k. \tag{35}$$

Substituting equation 35 into the upper-left block equation

$$\mathbf{A}_k \mathbf{E}_k + \mathbf{B}_k \mathbf{F}_k^\top = \mathbf{I}$$

gives

$$\left(\mathbf{A}_k - \mathbf{B}_k \mathbf{C}_k^{-1} \mathbf{B}_k^\top\right) \mathbf{E}_k = \mathbf{I}. \tag{36}$$

Therefore,

$$\mathbf{E}_k = \mathbf{S}_k^{-1}. \tag{37}$$

Since $\mathbf{H}^{(k)}$ is symmetric, $\mathbf{M}^{(k)}$ is also symmetric. Using equation 35 and equation 37, we obtain

$$\mathbf{F}_k = -\mathbf{E}_k \mathbf{B}_k \mathbf{C}_k^{-1}. \tag{38}$$

Equivalently,

$$\mathbf{B}_k \mathbf{C}_k^{-1} = -\mathbf{E}_k^{-1} \mathbf{F}_k. \tag{39}$$

Substituting equation 39 into equation 25 gives the inverse-based propagation formula

$$\boldsymbol{\delta}_{\mathcal{R}_k}^\star = -\mathbf{e}_k \mathbf{E}_k^{-1} \mathbf{F}_k. \tag{40}$$

After $\mathcal{G}_k$ has been quantized and its error has been propagated, the remaining active coordinates are $\mathcal{R}_k$. The Hessian block for the next step is therefore $\mathbf{C}_k$, so the next inverse Hessian should be $\mathbf{C}_k^{-1}$. From the block inverse identity,

$$\mathbf{C}_k^{-1} = \mathbf{G}_k - \mathbf{F}_k^\top \mathbf{E}_k^{-1} \mathbf{F}_k. \tag{41}$$

Thus the inverse carried to the next group step is

$$\mathbf{M}^{(k+1)} := \left(\mathbf{H}_{\mathcal{R}_k, \mathcal{R}_k}\right)^{-1} = \mathbf{G}_k - \mathbf{F}_k^\top \mathbf{E}_k^{-1} \mathbf{F}_k. \tag{42}$$

In implementation, we avoid explicitly forming $\mathbf{E}_k^{-1}$. Instead, we solve

$$\mathbf{E}_k \mathbf{Y}_k = \mathbf{F}_k \tag{43}$$

and use

$$\boldsymbol{\delta}_{\mathcal{R}_k}^\star = -\mathbf{e}_k \mathbf{Y}_k, \qquad \mathbf{M}^{(k+1)} = \mathbf{G}_k - \mathbf{F}_k^\top \mathbf{Y}_k. \tag{44}$$

For a full weight matrix, the current-group error is a matrix containing one row per output channel, and the same multiplier $\mathbf{Y}_k$ is applied on the right to all rows.

B.2 Derivation of Cross-Group Shrinkage

This appendix derives the shared cross-group shrinkage coefficient used in Section 5.2. The derivation first considers the optimal shrinkage for each cross-group Hessian entry, and then averages these entry-wise shrinkage factors to obtain the single layer-wise coefficient used by ReSC-Q.

Let $\hat{\mathbf{H}} = \mathbf{X}\mathbf{X}^\top$ be the empirical Hessian computed from N_{tok} calibration tokens. We work on the normalized covariance scale

$$\hat{\boldsymbol{\Sigma}} = \frac{\hat{\mathbf{H}}}{N_{\text{tok}}}, \tag{45}$$

so that the shrinkage coefficient is estimated independently of the number of tokens used to form $\hat{\mathbf{H}}$. Let $\boldsymbol{\Sigma}$ denote the corresponding true normalized covariance.

Let $\gamma(p)$ be the index of the quantization group containing coordinate p, and define the cross-group entry set

$$\mathcal{C} := \{(p, q) : \gamma(p) \neq \gamma(q)\}. \tag{46}$$

For any matrix $\mathbf{X}$, $\mathbf{X}_{\mathcal{C}}$ denotes the restriction of $\mathbf{X}$ to the entries in $\mathcal{C}$.

For estimating uncertainty, split the calibration tokens into K equal-sized chunks. Let $\hat{\mathbf{\Sigma}}^{(r)}$ be the normalized covariance estimate from chunk r, and define the chunk-averaged estimate

$$\bar{\mathbf{\Sigma}}_{\mathcal{C}} = \frac{1}{K} \sum_{r=1}^{K} \hat{\mathbf{\Sigma}}_{\mathcal{C}}^{(r)}. \tag{47}$$

For equal-sized chunks, this average matches the cross-group restriction of the full normalized covariance estimate. For each cross-group entry $(p, q) \in \mathcal{C}$, we use the noise model

$$\bar{\Sigma}_{pq} = \Sigma_{pq} + \Xi_{pq}, \qquad \mathbb{E}[\Xi_{pq}] = 0, \qquad V_{pq} := \mathbb{E}[\Xi_{pq}^2]. \tag{48}$$

For a fixed observed estimate $\bar{\Sigma}_{pq}$, ReSC-Q chooses an entry-wise shrinkage factor by minimizing

$$\mathcal{J}_{pq}(\rho) = \mathbb{E}\left[\left(\rho\bar{\Sigma}_{pq} - \Sigma_{pq}\right)^2\right] + \lambda_\rho \bar{\Sigma}_{pq}^2 \rho^2, \qquad 0 \leq \rho \leq 1. \tag{49}$$

The ridge term is scaled by the observed squared covariance magnitude $\bar{\Sigma}_{pq}^2$, so that the same λ_ρ can be used across entries with different scales.

Using $\bar{\Sigma}_{pq} = \Sigma_{pq} + \Xi_{pq}$, we have

$$\rho\bar{\Sigma}_{pq} - \Sigma_{pq} = (\rho - 1)\Sigma_{pq} + \rho\Xi_{pq}. \tag{50}$$

Since $\mathbb{E}[\Xi_{pq}] = 0$, the mixed term has zero expectation, and equation 49 becomes

$$\mathcal{J}_{pq}(\rho) = (1 - \rho)^2 \Sigma_{pq}^2 + \rho^2 V_{pq} + \lambda_\rho \bar{\Sigma}_{pq}^2 \rho^2. \tag{51}$$

Differentiating with respect to ρ gives

$$\frac{\partial \mathcal{J}_{pq}}{\partial \rho} = 2(\rho - 1)\Sigma_{pq}^2 + 2\rho\left(V_{pq} + \lambda_\rho \bar{\Sigma}_{pq}^2\right). \tag{52}$$

Setting the derivative to zero yields the entry-wise shrinkage factor

$$\rho_{pq}^\star = \frac{\Sigma_{pq}^2}{\Sigma_{pq}^2 + V_{pq} + \lambda_\rho \bar{\Sigma}_{pq}^2}. \tag{53}$$

Rather than applying a different coefficient to every cross-group entry, ReSC-Q uses a single shared coefficient obtained by averaging the entry-wise shrinkage factors:

$$\rho_{\mathcal{C}}^\star = \frac{1}{|\mathcal{C}|} \sum_{(p,q)\in\mathcal{C}} \rho_{pq}^\star. \tag{54}$$

Averaging over all cross-group entries makes the coefficient less sensitive to individual off-diagonal fluctuations while still reflecting the entry-wise uncertainty estimates.

We now describe the empirical plug-in estimator. The variance of the chunk mean for each entry is estimated by

$$\widehat{V}_{pq} = \frac{1}{K(K-1)} \sum_{r=1}^{K} \left(\hat{\Sigma}_{pq}^{(r)} - \bar{\Sigma}_{pq}\right)^2. \tag{55}$$

Under the independent chunk model, $\widehat{V}_{pq}$ estimates the variance of the averaged estimator $\bar{\Sigma}_{pq}$.

It remains to estimate the unknown squared covariance magnitude Σ_{pq}^2. From equation 48,

$$\mathbb{E}[\bar{\Sigma}_{pq}^2] = \Sigma_{pq}^2 + V_{pq}. \tag{56}$$

Thus, the observed squared estimate $\bar{\Sigma}_{pq}^2$ is upward-biased by the variance of the averaged estimator. We use the noise-corrected nonnegative estimate

$$\widehat{U}_{pq} := \max\left(\bar{\Sigma}_{pq}^2 - \widehat{V}_{pq}, 0\right). \tag{57}$$

Substituting $\widehat{U}_{pq}$ and $\widehat{V}_{pq}$ into equation 53 gives

$$\widehat{\rho}_{pq} = \frac{\widehat{U}_{pq}}{\widehat{U}_{pq} + \widehat{V}_{pq} + \lambda_\rho \bar{\Sigma}^2_{pq}}. \tag{58}$$

The shared shrinkage coefficient used by the layer is then

$$\widehat{\rho}_{\mathcal{C}} = \frac{1}{|\mathcal{C}|} \sum_{(p,q)\in\mathcal{C}} \widehat{\rho}_{pq}. \tag{59}$$

Finally, the coefficient estimated on the normalized covariance scale is applied to the empirical Hessian scale when constructing the propagation Hessian:

$$\tilde{H}_{pq} = \begin{cases} \hat{H}_{pq}, & \gamma(p) = \gamma(q), \\ \widehat{\rho}_{\mathcal{C}}\, \hat{H}_{pq}, & \gamma(p) \neq \gamma(q). \end{cases} \tag{60}$$

Thus, the within-group curvature is preserved, while only the cross-group interactions used for forward error compensation are shrunk.

B.3 From the Post-Compensation Residual to a Separable Upper Bound

This appendix derives the separable upper bound on the post-compensation residual used in Section 5.3. Different quantization representations can produce different current-group errors. We therefore derive the bound for an arbitrary row vector

$$\mathbf{u} \in \mathbb{R}^{1\times|\mathcal{G}_k|}$$

and later substitute the quantization error produced by the chosen quantizer.

From Appendix B.1, after the optimal future update, the residual local reconstruction loss for a candidate current-group error $\mathbf{e}_k$ is

$$\mathbf{e}_k \mathbf{S}_k \mathbf{e}_k^\top, \qquad \mathbf{S}_k = \mathbf{A}_k - \mathbf{B}_k \mathbf{C}_k^{-1} \mathbf{B}_k^\top. \tag{61}$$

The propagation Hessian is positive definite in practice, so $\mathbf{C}_k \succ \mathbf{0}$. Hence

$$\mathbf{B}_k \mathbf{C}_k^{-1} \mathbf{B}_k^\top \succeq \mathbf{0}. \tag{62}$$

Here $\preceq$ and $\succeq$ denote the Loewner order: for symmetric matrices $\mathbf{P}$ and $\mathbf{Q}$, $\mathbf{P} \preceq \mathbf{Q}$ means $\mathbf{Q} - \mathbf{P}$ is positive semidefinite. equation 62 implies

$$\mathbf{S}_k \preceq \mathbf{A}_k. \tag{63}$$

Therefore, for any current-group error vector $\mathbf{u}$,

$$\mathbf{u}\mathbf{S}_k\mathbf{u}^\top \leq \mathbf{u}\mathbf{A}_k\mathbf{u}^\top. \tag{64}$$

Replacing $\mathbf{S}_k$ by $\mathbf{A}_k$ is conservative; this remove the explicit future-compensation term $\mathbf{B}_k \mathbf{C}_k^{-1} \mathbf{B}_k^\top$, which is the part most directly affected by noisy cross-group estimates. Consider holding $\mathbf{A}_k$ and $\mathbf{C}_k$ fixed and replacing $\mathbf{B}_k$ with $\rho\mathbf{B}_k$. The resulting residual curvature is

$$\mathbf{S}_k(\rho) = \mathbf{A}_k - \rho^2 \mathbf{B}_k \mathbf{C}_k^{-1} \mathbf{B}_k^\top. \tag{65}$$

Since $\mathbf{B}_k \mathbf{C}_k^{-1} \mathbf{B}_k^\top \succeq \mathbf{0}$,

$$\mathbf{S}_k(1) \preceq \mathbf{S}_k(\rho) \preceq \mathbf{A}_k, \qquad 0 \leq \rho \leq 1. \tag{66}$$

Thus, cross-group shrinkage conservatively reduces the compensation benefit attributed to an uncertain coupling while preserving the within-group reconstruction objective and its upper-bound relation.

In contrast, shrinking the within-group off-diagonal entries replaces $\mathbf{A}_k$ with

$$\widetilde{\mathbf{A}}_k = \operatorname{diag}(\mathbf{A}_k) + \rho\left(\mathbf{A}_k - \operatorname{diag}(\mathbf{A}_k)\right). \tag{67}$$

Although $\widetilde{\mathbf{A}}_k$ remains positive semidefinite, $\widetilde{\mathbf{A}}_k - \mathbf{A}_k$ is generally indefinite. Therefore, neither $\widetilde{\mathbf{A}}_k \succeq \mathbf{A}_k$ nor $\widetilde{\mathbf{A}}_k \preceq \mathbf{A}_k$ is guaranteed, and a majorizer constructed from $\widetilde{\mathbf{A}}_k$ need not upper-bound

the original post-compensation residual. We therefore preserve $\mathbf{A}_k$ and selectively shrink only the uncertain cross-group coupling.

At the same time using only $\mathrm{diag}(\mathbf{A}_k)$ does not generally upper-bound the within-group quadratic form. For example, let

$$\mathbf{A} = \begin{bmatrix} 1 & c \\ c & 1 \end{bmatrix}, \qquad \mathbf{u} = [1 \quad 1], \qquad 0 < c < 1. \tag{68}$$

Then

$$\mathbf{u}\mathbf{A}\mathbf{u}^\top = 2 + 2c > 2 = \mathbf{u}\,\mathrm{diag}(\mathbf{A})\mathbf{u}^\top. \tag{69}$$

Hence, diagonal-only weighting can underestimate the original reconstruction loss for some error directions. We instead use the following row-sum loading, which provides a valid diagonal majorizer:

$$r_{k,j} = \sum_{\substack{t \in \mathcal{G}_k \\ t \neq j}} |(\mathbf{A}_k)_{jt}|, \qquad d_j = (\mathbf{A}_k)_{jj} + r_{k,j}, \tag{70}$$

and let

$$\mathbf{D}_{A,k} = \mathrm{diag}\{d_j : j \in \mathcal{G}_k\}. \tag{71}$$

We call $\mathbf{D}_{A,k}$ a diagonal majorizer of $\mathbf{A}_k$ because it upper-bounds the quadratic form of $\mathbf{A}_k$ for every row vector $\mathbf{u}$ supported on $\mathcal{G}_k$. Indeed,

$$\begin{aligned}
\mathbf{u}\mathbf{A}_k\mathbf{u}^\top &= \sum_{j \in \mathcal{G}_k} (\mathbf{A}_k)_{jj} u_j^2 + 2 \sum_{\substack{j<t \\ j,t \in \mathcal{G}_k}} (\mathbf{A}_k)_{jt} u_j u_t \\
&\leq \sum_{j \in \mathcal{G}_k} (\mathbf{A}_k)_{jj} u_j^2 + 2 \sum_{\substack{j<t \\ j,t \in \mathcal{G}_k}} |(\mathbf{A}_k)_{jt}|\,|u_j u_t| \\
&\leq \sum_{j \in \mathcal{G}_k} (\mathbf{A}_k)_{jj} u_j^2 + \sum_{\substack{j<t \\ j,t \in \mathcal{G}_k}} |(\mathbf{A}_k)_{jt}| \left(u_j^2 + u_t^2\right) \\
&= \sum_{j \in \mathcal{G}_k} \left((\mathbf{A}_k)_{jj} + \sum_{\substack{t \in \mathcal{G}_k \\ t \neq j}} |(\mathbf{A}_k)_{jt}| \right) u_j^2 \\
&= \sum_{j \in \mathcal{G}_k} d_j u_j^2 = \mathbf{u}\mathbf{D}_{A,k}\mathbf{u}^\top.
\end{aligned} \tag{72}$$

Combining equation 64 and equation 72, we obtain

$$\mathbf{u}\mathbf{S}_k\mathbf{u}^\top \leq \mathbf{u}\mathbf{A}_k\mathbf{u}^\top \leq \mathbf{u}\mathbf{D}_{A,k}\mathbf{u}^\top = \sum_{j \in \mathcal{G}_k} d_j u_j^2. \tag{73}$$

Finally, substitute the actual current-group quantization error

$$u_j = e_{k,j} = w_{k,j}^{\mathrm{eff}} - (sq_j - z). \tag{74}$$

Then

$$\mathbf{e}_k\mathbf{S}_k\mathbf{e}_k^\top \leq \sum_{j \in \mathcal{G}_k} d_j \left(w_{k,j}^{\mathrm{eff}} - (sq_j - z)\right)^2. \tag{75}$$

This gives the separable residual-aware objective used in equation 11. For the default affine implementation, substituting $\hat{w}_{k,j} = s \cdot q_j - z$ yields equation 12. The additional ridge penalty on s in the affine implementation is not part of the upper bound; it is introduced only for numerical stability when solving the regression.

B.4 Affine Quantizer Optimization under the Residual-Aware Objective

This appendix derives the closed-form affine-parameter updates for the group-wise affine implementation of the residual-aware objective in equation 12. We fix the current group $\mathcal{G}_k$ and a fixed integer assignment

$$Q = \{q_j\}_{j\in\mathcal{G}_k}, \qquad q_j \in \{0, \dots, 2^b - 1\}.$$

All sums below are over $j \in \mathcal{G}_k$. For readability, write

$$w_j := w_{k,j}^{\text{eff}}. \tag{76}$$

The weights d_j are the diagonal majorizer weights derived in Appendix B.3.

For fixed Q, the weighted regression problem is

$$\mathcal{L}(s, z; Q) = \sum_{j\in\mathcal{G}_k} d_j \left(w_j - (sq_j - z)\right)^2 + \lambda_s s^2. \tag{77}$$

Equivalently,

$$w_j - (sq_j - z) = w_j - sq_j + z.$$

Thus, conditional on Q, estimating (s, z) reduces to a weighted least-squares problem with an ℓ_2 penalty on the scale s.

Define the total weight and the weighted means

$$D_k = \sum_{j\in\mathcal{G}_k} d_j, \qquad \bar{w}_{\mathbf{d}} = \frac{1}{D_k}\sum_{j\in\mathcal{G}_k} d_j w_j, \qquad \bar{q}_{\mathbf{d}} = \frac{1}{D_k}\sum_{j\in\mathcal{G}_k} d_j q_j. \tag{78}$$

Since $d_j > 0$ for the positive-definite within-group curvature used in practice, $D_k > 0$.

For fixed s and Q, differentiating equation 77 with respect to z gives

$$\frac{\partial \mathcal{L}}{\partial z} = 2\sum_{j\in\mathcal{G}_k} d_j \left(w_j - sq_j + z\right). \tag{79}$$

Setting equation 79 to zero yields

$$\begin{aligned} 0 &= \sum_{j\in\mathcal{G}_k} d_j \left(w_j - sq_j + z\right) \\ &= \sum_{j\in\mathcal{G}_k} d_j w_j - s\sum_{j\in\mathcal{G}_k} d_j q_j + z\sum_{j\in\mathcal{G}_k} d_j. \end{aligned} \tag{80}$$

Solving for z gives

$$\begin{aligned} \widehat{z}(s) &= \frac{s\sum_{j\in\mathcal{G}_k} d_j q_j - \sum_{j\in\mathcal{G}_k} d_j w_j}{\sum_{j\in\mathcal{G}_k} d_j} \\ &= s\bar{q}_{\mathbf{d}} - \bar{w}_{\mathbf{d}}. \end{aligned} \tag{81}$$

Substituting equation 81 into the reconstruction error gives the centered residual

$$\begin{aligned} w_j - (sq_j - \widehat{z}(s)) &= w_j - sq_j + s\bar{q}_{\mathbf{d}} - \bar{w}_{\mathbf{d}} \\ &= (w_j - \bar{w}_{\mathbf{d}}) - s(q_j - \bar{q}_{\mathbf{d}}). \end{aligned} \tag{82}$$

Therefore, after optimizing over the offset, the problem reduces to a one-dimensional ridge regression in s:

$$\mathcal{L}(s; Q) = \sum_{j\in\mathcal{G}_k} d_j \left[(w_j - \bar{w}_{\mathbf{d}}) - s(q_j - \bar{q}_{\mathbf{d}})\right]^2 + \lambda_s s^2. \tag{83}$$

Differentiating equation 83 with respect to s gives

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial s} &= -2\sum_{j\in\mathcal{G}_k} d_j(q_j - \bar{q}_{\mathbf{d}}) \left[(w_j - \bar{w}_{\mathbf{d}}) - s(q_j - \bar{q}_{\mathbf{d}})\right] + 2\lambda_s s \\ &= -2\sum_{j\in\mathcal{G}_k} d_j(q_j - \bar{q}_{\mathbf{d}})(w_j - \bar{w}_{\mathbf{d}}) + 2s\sum_{j\in\mathcal{G}_k} d_j(q_j - \bar{q}_{\mathbf{d}})^2 + 2\lambda_s s. \end{aligned} \tag{84}$$

Setting equation 84 to zero yields

$$\widehat{s} = \frac{\sum_{j\in\mathcal{G}_k} d_j(q_j - \bar{q}_{\mathbf{d}})(w_j - \bar{w}_{\mathbf{d}})}{\sum_{j\in\mathcal{G}_k} d_j(q_j - \bar{q}_{\mathbf{d}})^2 + \lambda_s}. \tag{85}$$

The corresponding optimal offset is

$$\widehat{z} = \widehat{z}(\widehat{s}) = \widehat{s}\bar{q}_{\mathbf{d}} - \bar{w}_{\mathbf{d}}. \tag{86}$$

It is convenient to express equation 85 using unnormalized weighted moments:

$$\mathrm{Cov}_{\mathbf{d}}(W_k^{\mathrm{eff}}, Q) := \sum_{j\in\mathcal{G}_k} d_j(q_j - \bar{q}_{\mathbf{d}})(w_{k,j}^{\mathrm{eff}} - \bar{w}_{\mathbf{d}}), \tag{87}$$

$$\mathrm{Var}_{\mathbf{d}}(Q) := \sum_{j\in\mathcal{G}_k} d_j(q_j - \bar{q}_{\mathbf{d}})^2. \tag{88}$$

With these definitions,

$$\widehat{s} = \frac{\mathrm{Cov}_{\mathbf{d}}(W_k^{\mathrm{eff}}, Q)}{\mathrm{Var}_{\mathbf{d}}(Q) + \lambda_s}, \qquad \widehat{z} = \widehat{s}\bar{q}_{\mathbf{d}} - \bar{w}_{\mathbf{d}}. \tag{89}$$

This matches the closed-form update used in the main text.

Algorithm 1 Layer-wise ReSC-Q

Input: Pre-trained model $\mathcal{M}$ with L layers, calibration data X, bit-width b, iterations T, shrinkage ridge λ_ρ, regression ridge λ_s
Output: Quantized model $\hat{\mathcal{M}}$

for $l = 1$ to L **do**
  $X^{(l)} \leftarrow$ accumulated input activations of layer l
  Estimate $\hat{\mathbf{H}}, \widehat{\mathbf{V}}_\Sigma, N$ from $X^{(l)}$
  Partition channels into m groups $\{\mathcal{G}_1, ..., \mathcal{G}_m\}$
  # Find Cross-Group Shrinkage Coefficient
  $\rho \leftarrow$ EstimateShrinkage($\hat{\mathbf{H}}, \widehat{\mathbf{V}}_\Sigma, N, \lambda_\rho$)
  Build shrunk $\tilde{\mathbf{H}}$ from $\hat{\mathbf{H}}$ and $\hat{\rho}$ using Eq. 9
  $\mathbf{W}_{\mathrm{work}} \leftarrow \mathbf{W}^{(l)}, \quad \mathbf{M} \leftarrow \tilde{\mathbf{H}}^{-1}, \quad \mathcal{Q}^{(l)} \leftarrow [\,]$
  for $k = 1$ to m **do**
    $\mathbf{w}^{\mathrm{eff}} \leftarrow \mathbf{W}_{\mathrm{work}}[:, \mathcal{G}_k], \quad \mathcal{R}_k \leftarrow \bigcup_{r>k} \mathcal{G}_r$
    # Quantize Current Weight Block
    $(Q, s, z) \leftarrow$ Quantize($\mathbf{w}^{\mathrm{eff}}, \tilde{\mathbf{H}}_{\mathcal{G}_k}, ...$)
    $\hat{\mathbf{w}} \leftarrow sQ - z, \qquad \mathbf{e} \leftarrow \mathbf{w}^{\mathrm{eff}} - \hat{\mathbf{w}}$
    Append $\{Q, s, z\}$ to $\mathcal{Q}^{(l)}$
    # Blockwise Error Compensation
    if $k < m$ **then**
      $(\delta, \mathbf{M}) \leftarrow$ Propagate($\mathbf{e}, \mathbf{M}, |\mathcal{G}_k|$)
      $\mathbf{W}_{\mathrm{work}}[:, \mathcal{R}_k] \leftarrow \mathbf{W}_{\mathrm{work}}[:, \mathcal{R}_k] + \delta$
  Construct $\hat{\mathcal{M}}^{(l)}$ from $\mathcal{Q}^{(l)}$
  $X^{(l+1)} \leftarrow \hat{\mathcal{M}}^{(l)}(X^{(l)})$
return $\hat{\mathcal{M}}$

procedure EstimateShrinkage($\hat{\mathbf{H}}, \dots$) ▷ §5.2
  $\hat{\mathbf{\Sigma}} \leftarrow \hat{\mathbf{H}}/N, \quad \rho_{\mathrm{sum}} \leftarrow 0, \quad n_C \leftarrow 0$
  for (p, q) with $\gamma(p) \neq \gamma(q)$ **do**
    $\widehat{V}_{pq} \leftarrow (\widehat{\mathbf{V}}_\Sigma)_{pq}, \ \widehat{U}_{pq} \leftarrow \max(\hat{\Sigma}_{pq}^2 - \widehat{V}_{pq}, 0)$
    $\widehat{\rho}_{pq} \leftarrow \widehat{U}_{pq} / \big(\widehat{U}_{pq} + \widehat{V}_{pq} + \lambda_\rho \hat{\Sigma}_{pq}^2\big)$ ▷ Eq. 58
    $\rho_{\mathrm{sum}} \leftarrow \rho_{\mathrm{sum}} + \widehat{\rho}_{pq}, \quad n_C \leftarrow n_C + 1$
  $\rho \leftarrow \rho_{\mathrm{sum}}/n_C$
  return ρ

procedure Propagate($\mathbf{e}, \mathbf{M}, n_g$) ▷ §5.1
  $\mathbf{M} = \begin{bmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{F}^\top & \mathbf{G} \end{bmatrix}$ such that $\mathbf{E} \in \mathbb{R}^{n_g \times n_g}$
  Solve $\mathbf{E}\mathbf{Y} = \mathbf{F}$ via Cholesky factorization
  $\delta \leftarrow -\mathbf{e}\mathbf{Y}, \quad \mathbf{M} \leftarrow \mathbf{G} - \mathbf{F}^\top\mathbf{Y}$ ▷ Eq. 42
  return $\delta, \mathbf{M}$

procedure Quantize($\mathbf{w}^{\mathrm{eff}}, \mathbf{A}, T, b, \lambda_s$) ▷ §5.3
  # Residual-Aware Quantization
  $d_j \leftarrow \mathbf{A}_{jj} + \sum_{t\neq j} |\mathbf{A}_{jt}|$
  Initialize s, z from $\mathbf{w}^{\mathrm{eff}}$ using Eq. 1
  for $t = 0$ to $T - 1$ **do**
    $Q \leftarrow \mathrm{clip}\big(\lfloor (\mathbf{w}^{\mathrm{eff}} + z)/s \rceil, \ 0, \ 2^b - 1\big)$
    $s \leftarrow \mathrm{Cov}_{\mathbf{d}}(\mathbf{w}^{\mathrm{eff}}, Q) / \big(\mathrm{Var}_{\mathbf{d}}(Q) + \lambda_s\big)$
    $z \leftarrow s\bar{q}_{\mathbf{d}} - \bar{w}_{\mathbf{d}}$ ▷ $\bar{q}_{\mathbf{d}}, \bar{w}_{\mathbf{d}}$ from Eq. 78
  return Q, s, z

C Algorithmic Summary

Algorithm 1 summarizes the layer-wise implementation of ReSC-Q. For each layer, we first collect its calibration activations and estimate the empirical Hessian together with the covariance uncertainty statistics. The procedure EstimateShrinkage computes the shared cross-group coefficient on the normalized covariance scale, which is then used to construct the propagation Hessian and initialize its inverse.

Quantization proceeds sequentially over the affine quantization groups. For each group, Quantize constructs the row-sum majorizer from the preserved within-group curvature and alternates integer refinement with weighted regression of the shared scale and zero-point. The resulting group error is

then passed to PROPAGATE, which updates the remaining weights and the active inverse-Hessian state. Thus, residual-aware current-group quantization and cross-group error compensation are implemented as separate operations while sharing the same block partition.

The algorithm is written in matrix form: the current block $\mathbf{w}^{\text{eff}}$ contains all output rows restricted to $\mathcal{G}_k$, whereas (s, z) are estimated independently for each output row and shared across the coordinates within the group. After all groups are processed, the layer is reconstructed from $\{Q, s, z\}$ and used to generate the calibration activations for the next layer. The Hessian, shrinkage coefficient, and inverse-Hessian state are required only during offline quantization and are not part of the final quantized representation.

D ADDITIONAL EVALUATION

For completeness, we collect several supplementary results that are useful for interpreting the main evaluation but were omitted from the main text for space.

Table 4: Large-model results at 2-bit. We report WikiText-2 perplexity (PPL), zero-shot task accuracies, and quantized model size.

Method	W-bit	PPL ↓	Zero-Shot Reasoning Tasks									Model Size (GB)
			ARC-C	ARC-E	BoolQ	Hella	OBQA	PIQA	SIQA	Wino	Avg ↑	
Llama-3.1-70B-Instruct												
Baseline	16	3.78	62.37	83.46	89.08	85.28	47.80	83.84	50.92	83.11	73.23	131.42
GPTQ	2	168.49	26.79	25.46	37.83	26.15	29.60	51.90	34.65	49.25	35.20	27.82
GPTQ+QEP	2	-	25.26	27.10	37.83	25.40	27.00	50.60	34.95	48.30	34.56	27.82
OWQ	2.16	9.94	45.39	68.73	71.13	70.94	38.40	76.28	35.47	68.11	59.31	29.17
DASH-Q	2	7.97	56.14	79.08	85.87	77.70	**44.20**	**81.50**	40.33	74.90	67.47	27.82
ReSC-Q	2	**7.28**	**58.45**	**81.52**	**86.82**	**77.85**	42.40	81.45	**42.48**	**79.32**	**68.79**	27.82
GLM-4.5-Air												
Baseline	16	4.87	62.03	82.66	88.13	85.85	49.20	83.62	46.62	77.35	71.93	199.03
GPTQ	2	94.48	28.58	46.46	61.31	48.45	28.20	60.39	32.75	55.33	45.18	39.23
GPTQ+QEP	2	52.94	41.89	57.58	63.67	67.32	36.60	72.25	33.42	61.56	54.29	39.23
OWQ	2.65	7.55	53.33	74.12	80.09	79.39	44.00	79.87	40.99	74.19	65.75	47.42
DASH-Q	2	7.46	59.30	82.87	85.69	**81.84**	**47.80**	**83.57**	44.52	74.35	69.99	39.23
ReSC-Q	2	**7.43**	**60.49**	**84.60**	**87.19**	81.34	47.20	82.86	**44.63**	**74.74**	**70.38**	39.23
Qwen3.5-122B-A10B												
Baseline	16	5.07	63.91	80.26	86.33	85.96	46.80	83.68	46.11	75.37	71.05	227.45
GPTQ	2	-	22.70	25.08	37.83	25.04	27.60	49.51	32.91	49.57	33.78	45.02
GPTQ+QEP	2	8.01	57.34	75.72	83.64	77.76	40.00	79.16	40.89	67.88	65.30	45.02
OWQ	2.89	7.14	57.17	78.75	72.81	79.74	42.20	79.43	42.37	71.19	65.46	57.90
DASH-Q	2	8.05	61.77	81.36	88.04	79.22	44.80	81.01	42.78	70.01	68.62	45.02
ReSC-Q	2	**6.87**	**63.40**	**84.30**	**88.65**	**80.74**	**45.40**	**81.34**	**44.37**	**72.30**	**70.06**	45.02
Mistral-Large-Instruct-2407												
Baseline	16	2.74	65.44	83.33	90.43	87.72	51.80	85.42	47.80	82.24	74.27	228.38
GPTQ	2	4.94	51.45	71.93	86.54	79.57	36.40	79.22	37.05	64.80	63.37	44.04
GPTQ+QEP	2	5.20	51.96	70.24	85.26	80.52	37.20	79.49	39.25	68.11	64.01	44.04
OWQ	2.12	4.80	54.44	76.56	87.55	82.04	45.80	81.61	45.24	79.01	69.03	45.75
DASH-Q	2	4.43	58.70	77.90	89.11	83.14	**49.00**	**83.46**	44.98	80.58	70.86	44.04
ReSC-Q	2	**4.14**	**63.40**	**82.11**	**89.42**	**83.88**	48.60	**83.46**	**45.75**	**80.82**	**72.18**	44.04

D.1 Additional Large-Model Results

Table 4 reports the full task-wise results corresponding to the large-model comparison in Table 2. We include the BF16 reference together with GPTQ, GPTQ+QEP, OWQ, DASH-Q, and ReSC-Q. Across the four evaluated models, ReSC-Q achieves the lowest perplexity and highest average zero-shot accuracy among the compared quantized methods. The task-wise results show that these improvements are generally distributed across multiple reasoning tasks rather than being driven by a single benchmark.

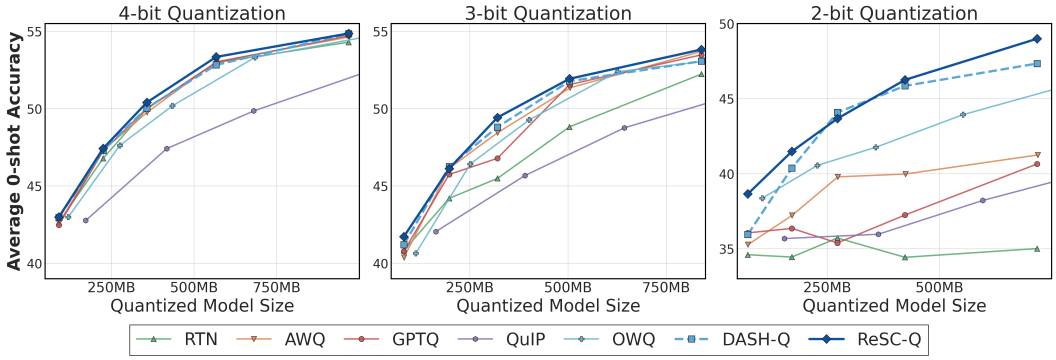

Figure 4: Average zero-shot accuracy versus quantized model size on the `MobileLLM` family.

D.2 Pareto Frontier Trade-off on Smaller Model

Fig. 4 plots average zero-shot accuracy against the final quantized model size on the MobileLLM family (`125M`, `350M`, `600M`, `1B`, and `1.5B`). At 4-bit, most methods achieve similar accuracy and the frontier is relatively tight. The separation becomes clearer at 3-bit and is largest at 2-bit, where ReSC-Q generally achieves the highest accuracy at a comparable model size.

Table 5: Sensitivity to calibration size on `Llama-3.1-8B-Instruct` at 2-bit. We report WikiText-2 perplexity and average zero-shot accuracy. ReSC-Q$_{\rho=1}$ uses the same residual-aware quantization and blockwise compensation as ReSC-Q with the shrinkage coefficient fixed to $\rho = 1$.

	GPTQ		DASH-Q		ReSC-Q$_{\rho=1}$		ReSC-Q	
n	PPL	Zero-shot	PPL	Zero-shot	PPL	Zero-shot	PPL	Zero-shot
2	93607.17	34.73	16.89	57.19	24.87	42.57	13.73	57.20
4	5528.60	35.52	17.34	56.77	18.84	47.75	13.34	56.57
8	895.71	35.43	16.98	56.64	15.02	51.85	12.89	58.52
16	99.03	37.98	17.07	56.78	13.10	53.65	12.54	57.89
32	35.44	35.25	16.50	56.72	12.38	56.29	12.28	58.50
64	34.46	38.00	16.65	56.51	11.93	57.18	12.09	58.87
128	29.39	37.89	17.17	56.69	11.58	56.91	11.76	58.80
256	25.53	35.17	16.82	56.56	11.57	56.27	11.96	58.36
512	24.24	35.11	16.81	56.47	11.61	58.12	11.58	59.45
1024	26.14	35.71	16.96	56.39	11.49	57.83	11.55	59.55

D.3 Sensitivity to Calibration Sample Size

Table 5 reports the exact values corresponding to Fig. 3. We compare GPTQ, DASH-Q, ReSC-Q$_{\rho=1}$, and ReSC-Q over a wide range of calibration sizes. ReSC-Q$_{\rho=1}$ uses the same residual-aware within-group objective and blockwise error compensation as ReSC-Q, but fixes the cross-group shrinkage coefficient to $\rho = 1$. Thus, the comparison between ReSC-Q$_{\rho=1}$ and ReSC-Q directly isolates the effect of cross-group shrinkage.

At small calibration sizes, GPTQ exhibits severe degradation in both perplexity and zero-shot accuracy. DASH-Q is substantially less sensitive to the calibration budget, providing a reference that does not use cross-group error compensation. ReSC-Q$_{\rho=1}$ also degrades at small calibration sizes despite using the residual-aware within-group objective, showing that this objective alone does not prevent unreliable cross-group compensation. In contrast, ReSC-Q substantially reduces this degradation by shrinking the cross-group estimates before they are used for compensation.

As the calibration size increases, the gap between ReSC-Q$_{\rho=1}$ and ReSC-Q becomes smaller in perplexity, consistent with the increasing estimated shrinkage coefficients reported in Table 3. At the same time, ReSC-Q maintains consistently stronger zero-shot accuracy than the unshrunk variant. These results indicate that aggressive shrinkage is most important when cross-group statistics are estimated from limited data, while more of the estimated correlation can be retained as the calibration budget increases.

Table 6: Ablation of the within-group surrogate and shrinkage scope on Llama-3.1-8B-Instruct at 2-bit. Each calibration sample contains 2,048 tokens. $\mathbf{A}_{jj}$ uses only the diagonal entries of the within-group curvature; $\widetilde{\mathbf{A}}$ applies shrinkage to all off-diagonal entries, including within-group entries; $\mathbf{d}_j$(Ours) preserves the within-group curvature and shrinks only cross-group entries.

	Perplexity ↓			**Zero-Shot Average ↑**		
n	$\mathbf{A}_{jj}$	$\widetilde{\mathbf{A}}$	d_j (Ours)	$\mathbf{A}_{jj}$	$\widetilde{\mathbf{A}}_k$	d_j (Ours)
4	13.77	13.36	**13.34**	56.90	**57.20**	56.57
8	12.99	13.03	**12.89**	58.30	**58.60**	58.52
16	12.70	12.64	**12.54**	**58.60**	58.30	57.89
32	**12.20**	12.53	12.28	58.20	57.40	**58.50**
64	12.36	12.38	**12.09**	58.10	58.40	**58.87**
128	11.88	12.12	**11.76**	**59.00**	58.10	58.80

D.4 Ablation of Shrinkage Scope and Within-Group Surrogate

We isolate two design choices in ReSC-Q: how the preserved within-group curvature is converted into a separable quantization objective, and whether shrinkage is restricted to cross-group terms. For the first, the A_{jj} variant replaces the row-sum majorizer d_j with the diagonal weight $(\mathbf{A}_k)_{jj}$ while retaining the same cross-group shrinkage and blockwise compensation. For the second, the $\widetilde{\mathbf{A}}$ variant additionally applies the estimated coefficient $\hat{\rho}$ to the within-group off-diagonal entries. ReSC-Q uses the row-sum majorizer d_j and restricts shrinkage to the cross-group blocks used for error compensation.

Table 6 shows that ReSC-Q achieves the lowest perplexity in most calibration settings. Applying shrinkage uniformly to all off-diagonal entries occasionally improves an individual zero-shot result when calibration data are scarce, but does not provide a consistent benefit and also modifies the within-group fitting objective. Similarly, the A_{jj} variant can improve an individual metric, but discards the within-group correlations required for the residual upper bound.

Overall, preserving the within-group curvature through the row-sum majorizer while restricting shrinkage to cross-group couplings gives the most consistent reconstruction-oriented performance, achieving the lowest perplexity in most settings. This behavior is consistent with Appendix B.3: cross-group shrinkage regularizes the curvature used for compensation without altering $\mathbf{A}_k$, while the row-sum majorizer provides a separable upper bound on the post-compensation residual. The variation in zero-shot accuracy also reflects that improvements in the local reconstruction objective need not translate monotonically to downstream task accuracy.

D.5 Comparison with Sensitivity-Weighted Quantizers

We additionally evaluate SqueezeLLM,(8) and LeanQuant,(9), which use per-weight sensitivity to construct non-uniform quantizers. Because their native representations differ from the group-wise affine representation of ReSC-Q, we report them as broader low-bit references rather than strictly format-matched baselines.

Table 7: Comparison with sensitivity-weighted quantizers on Llama-3.1-8B-Instruct under 2-bit quantization with $n = 128$ calibration samples, each containing 2,048 tokens. SqueezeLLM and LeanQuant use their native non-uniform quantizers, whereas the ReSC-Q rows use the group-wise affine representation with group size of 32. The SqueezeLLM row reports only its dense quantizer and excludes its FP16 sparse component. We report WikiText-2 perplexity, zero-shot task accuracies.

Method	Weighting / Quantizer	PPL ↓	ARC-C	ARC-E	BoolQ	Hella	OBQA	PIQA	SIQA	Wino	Avg ↑
SqueezeLLM	Fisher-weighted k-means	386.75	28.84	47.77	40.24	37.93	27.40	61.32	34.19	51.46	41.14
LeanQuant	$a_j^{(4)}$-weighted k-means	22.31	27.30	47.73	68.47	43.47	28.80	60.88	34.24	59.04	46.24
DASH-Q	Hessian Diagonal A_{jj}	17.17	39.25	67.05	76.15	62.09	34.60	73.45	33.98	**66.93**	56.69
DASH-Q	Fisher $f_{i,j}$	**15.14**	**40.87**	**68.52**	**77.49**	**65.31**	**35.40**	**74.65**	**35.41**	66.38	**58.00**
DASH-Q	Hessian Inverse $a_j^{(0)}$	1825.74	23.38	30.22	49.57	29.55	24.80	55.93	34.85	50.36	37.33
DASH-Q	Hessian Inverse $a_j^{(1)}$	43.31	27.39	45.33	64.01	45.99	29.20	64.85	34.39	57.62	46.10
DASH-Q	Hessian Inverse $a_j^{(2)}$	30.22	29.52	48.27	68.56	49.86	29.60	66.27	33.73	59.91	48.21
DASH-Q	Hessian Inverse $a_j^{(3)}$	26.89	29.95	46.72	55.35	50.39	29.40	67.08	33.42	59.75	46.51
DASH-Q	LeanQuant $a_j^{(4)}$	33.51	29.95	45.33	38.90	47.80	30.20	65.89	33.32	58.01	43.67
ReSC-Q	Row-sum Majorizer d_j	11.76	**41.38**	**71.42**	80.24	64.30	**36.80**	73.99	34.90	67.56	**58.83**
ReSC-Q	Fisher $f_{i,j}$	**11.51**	40.36	70.08	76.27	**65.66**	36.60	**75.57**	**35.52**	67.56	58.45
ReSC-Q	Hessian Inverse $a_j^{(0)}$	16.04	23.98	37.84	56.48	33.62	26.20	60.39	34.85	53.75	40.89
ReSC-Q	Hessian Inverse $a_j^{(1)}$	12.10	39.33	68.18	**81.07**	62.90	35.80	72.96	35.47	**69.61**	58.17
ReSC-Q	Hessian Inverse $a_j^{(2)}$	12.32	35.92	61.62	79.42	61.78	34.80	73.94	35.06	67.96	56.31
ReSC-Q	Hessian Inverse $a_j^{(3)}$	12.53	36.18	61.78	77.09	60.04	36.40	73.39	34.54	67.17	55.83
ReSC-Q	LeanQuant $a_j^{(4)}$	13.44	37.54	64.60	78.81	57.64	36.60	71.76	34.19	65.51	55.83

SqueezeLLM combines empirical-Fisher-weighted k-means with a dense-and-sparse decomposition that retains selected weights in FP16. To isolate its dense non-uniform quantizer without introducing additional high-precision sparse storage, we disable the sparse component and run weighted k-means for 50 iterations. LeanQuant learns a reconstruction-loss-aware non-uniform grid using

$$a_j^{(p)} = \left[(\mathbf{H}^{-1})_{jj}\right]^{-p}. \tag{90}$$

For its native quantizer, we follow the original setting with $p = 4$, run k-means for 100 iterations, and apply the uniformly spaced centroid initialization proposed by LeanQuant.

To isolate the effect of the sensitivity signal from that of the non-uniform codebook, we additionally evaluate the same importance weights under group-wise affine quantization. For a given within-group weight $\omega_{i,j}$, we solve

$$\min_{s,z,Q} \sum_{j \in \mathcal{G}_k} \omega_{i,j} \left(w_{i,j}^{\text{eff}} - (sq_{i,j} - z)\right)^2 + \lambda_s s^2, \tag{91}$$

where $\omega_{i,j}$ is replaced by the empirical-Fisher weight $f_{i,j}$ or the inverse-Hessian weight $a_j^{(p)}$. We evaluate these weights under two affine settings: DASH-Q, whose default weighting uses the Hessian diagonal $(\mathbf{A}_k)_{jj}$, and ReSC-Q, whose default weighting uses the row-sum majorizer d_j. For each setting, all components other than the within-group fitting weight are kept unchanged. In particular, ReSC-Q retains its cross-group shrinkage and blockwise error compensation. We sweep $p \in \{0, 1, 2, 3, 4\}$ for the transferred inverse-Hessian weight.

Table 7 reports the native non-uniform quantizers together with the controlled affine weighting ablations. Among the native non-uniform methods, LeanQuant substantially improves perplexity over the dense-only SqueezeLLM result, while both remain below the affine ReSC-Q variants in this setting.

Within the controlled affine settings, empirical-Fisher weighting improves DASH-Q in both perplexity and average zero-shot accuracy. Under ReSC-Q, however, it yields slightly lower perplexity but also slightly lower average zero-shot accuracy than the row-sum majorizer. Thus, the benefit of an alternative sensitivity signal depends on the surrounding quantization and compensation objective, and Fisher weighting does not consistently improve the complete ReSC-Q pipeline.

The transferred inverse-Hessian weighting is also sensitive to its exponent. Reducing p from the original LeanQuant setting of 4 substantially improves the affine results, with $p = 1$ giving the best

overall trade-off among the tested exponents. Nevertheless, it remains below the row-sum majorizer in both perplexity and average zero-shot accuracy under ReSC-Q. Together with the Fisher results, this indicates that substituting a generic sensitivity score for d_j does not consistently improve the residual-aware objective.

Table 8: Effect of group size at 2-bit. We report perplexity and average zero-shot accuracy for DASH-Q and ReSC-Q at group sizes 32, 64, and 128.

		Perplexity						Average Zero-Shot Accuracy					
		DASH-Q			ReSC-Q			DASH-Q			ReSC-Q		
Model	GS	32	64	128	32	64	128	32	64	128	32	64	128
Llama-3.1-8B		17.17	25.61	61.10	11.76	13.88	16.15	56.69	49.82	44.49	58.80	55.89	53.33
Qwen3-14B		11.41	12.57	14.58	10.79	11.37	11.89	64.03	61.15	58.23	64.92	61.66	58.97
Mixtral-8x7B		5.51	5.85	6.32	5.22	5.46	5.64	69.77	63.95	59.71	69.93	68.66	68.01

D.6 Sensitivity to Group Size at 2-bit

Table 8 reports the effect of group size at 2-bit. As the group size increases, a single affine quantizer must represent a larger and more heterogeneous set of weights, making the within-group quantization problem more difficult. Accordingly, perplexity increases and average zero-shot accuracy decreases for both DASH-Q and ReSC-Q across all three models.

ReSC-Q degrades more gradually as the group size increases, particularly on `Llama-3.1-8B` and `Mixtral-8x7B`. This trend is consistent with its residual-aware formulation: larger groups can produce larger current-group quantization errors, while blockwise compensation allows part of the resulting reconstruction error to be absorbed by later groups. Cross-group shrinkage controls how strongly the estimated correlations are used for this compensation. These results support the smaller group sizes used in the main experiments while showing that ReSC-Q remains effective when wider quantization groups are used.

Table 9: Compatibility of QEP with GPTQ and ReSC-Q under 2-bit quantization with group size 32. The coefficient γ denotes QEP's `perccorr` parameter. We report WikiText-2 perplexity and zero-shot task accuracies.

Method	γ	**PPL** ↓	**Zero-Shot Reasoning Tasks**								**Avg** ↑
			ARC-C	**ARC-E**	**BoolQ**	**Hella**	**OBQA**	**PIQA**	**SIQA**	**Wino**	
Llama-3.1-8B-Instruct											
Baseline (BF16)	–	7.22	55.2	79.7	85.4	79.5	45.0	81.1	42.3	73.9	67.76
GPTQ	–	28.33	25.9	26.5	38.2	26.0	30.6	54.0	34.5	49.6	35.66
GPTQ + QEP	0.5	15.77	26.0	25.1	43.2	26.3	28.4	51.8	34.6	49.5	35.61
ReSC-Q	–	11.76	**41.4**	**71.4**	**80.2**	**64.3**	**36.8**	74.0	**34.9**	67.6	**58.83**
ReSC-Q + QEP	0.5	13.16	32.6	57.2	70.9	53.2	32.2	68.0	33.6	63.9	51.43
ReSC-Q + QEP	0.25	**11.06**	38.7	68.3	80.1	62.8	36.0	73.2	33.6	**68.1**	57.60
ReSC-Q + QEP	0.1	11.46	40.7	69.4	79.2	63.3	35.4	**74.6**	34.5	68.0	58.14
Qwen3-14B											
Baseline (BF16)	–	8.65	60.4	82.8	89.3	78.8	46.2	80.0	44.3	72.6	69.30
GPTQ	–	13.65	33.7	52.1	72.4	59.3	36.2	68.7	35.4	56.4	51.76
GPTQ + QEP	0.5	13.88	40.0	60.4	79.0	61.4	35.4	69.2	36.3	59.8	55.20
ReSC-Q	–	10.79	55.1	79.1	87.4	67.8	**42.6**	76.2	40.6	**70.6**	**64.92**
ReSC-Q + QEP	0.5	12.38	49.8	77.1	82.4	64.1	39.0	74.4	38.3	67.3	61.54
ReSC-Q + QEP	0.25	10.94	**55.7**	**79.4**	87.9	67.6	41.0	76.0	**40.8**	69.5	64.72
ReSC-Q + QEP	0.1	**10.73**	54.1	78.9	**88.0**	**68.3**	41.0	**76.6**	39.8	70.2	64.60

D.7 Compatibility with Cross-Layer Error Correction

We examine whether ReSC-Q is compatible with QEP (19). QEP is an add-on to sequential layer-wise PTQ that explicitly accounts for quantization errors accumulated in previous layers. At each layer, QEP estimates the mismatch between the full-precision and quantized upstream activations and uses

it to shift the current weight target before quantization. Its `perccorr` coefficient γ controls the correction strength: $\gamma = 0$ recovers the original layer-wise PTQ objective, whereas larger values apply a greater fraction of the estimated cross-layer correction.

We evaluate QEP with GPTQ and ReSC-Q under 2-bit quantization with group size 32 on Llama-3.1-8B-Instruct and Qwen3-14B. Following its recommended configuration, we use $\gamma = 0.5$ for GPTQ. For ReSC-Q, we additionally evaluate $\gamma \in \{0.5, 0.25, 0.1\}$ to examine whether the two methods are composable.

As shown in Table 9, QEP is effective when the base quantizer leaves substantial error to be propagated across layers. On Llama-3.1-8B, it reduces GPTQ perplexity from 28.33 to 15.77, while on Qwen3-14B it improves average zero-shot accuracy from 51.76 to 55.20. In contrast, applying the same $\gamma = 0.5$ correction to ReSC-Q degrades performance on both models. This suggests that, when the base quantizer already significantly reduces within-layer residual error, a strong cross-layer correction may over-adjust the current target and become more sensitive to estimation noise. Reducing γ recovers the performance: $\gamma = 0.25$ improves Llama-3.1-8B perplexity from 11.76 to 11.06, while smaller coefficients remain comparable to ReSC-Q on Qwen3-14B. These results show that cross-layer correction can be combined with ReSC-Q, but requires a more conservative correction strength; in our experiments, smaller γ can improve perplexity while preserving the zero-shot accuracy of ReSC-Q.

Table 10: Compatibility of ReSC-Q with rotation-based quantization under 2-bit precision. We report WikiText-2 perplexity and zero-shot task accuracies at group sizes 32 and 128.

Method	**PPL** ↓	**Zero-Shot Reasoning Tasks**								**Avg** ↑
		ARC-C	**ARC-E**	**BoolQ**	**Hella**	**OBQA**	**PIQA**	**SIQA**	**Wino**	
Llama-3.1-8B-Instruct (gs: 32)										
QuIP	21.20	26.0	36.1	50.1	37.0	28.0	57.1	33.3	51.6	39.90
QuaRot	20.89	28.8	35.0	54.0	45.2	28.0	61.6	34.5	54.5	42.68
QuaRot + ReSC-Q	**11.32**	**42.9**	68.6	78.8	**64.6**	35.4	**74.9**	34.3	**69.5**	58.64
ReSC-Q	11.76	41.4	**71.4**	**80.2**	64.3	**36.8**	74.0	**34.9**	67.6	**58.83**
Qwen3-14B (gs: 32)										
QuIP	11.81	36.0	54.3	71.5	64.9	35.6	70.5	37.4	60.3	53.81
QuaRot	11.79	37.3	58.5	67.7	64.8	35.4	71.7	36.1	60.1	53.94
QuaRot + ReSC-Q	10.92	52.9	77.7	87.0	**69.7**	42.2	74.6	39.6	70.2	64.23
ReSC-Q	**10.79**	**55.1**	**79.1**	**87.4**	67.8	**42.6**	**76.2**	**40.6**	**70.6**	**64.92**
Llama-3.1-8B-Instruct (gs: 128)										
QuaRot + ReSC-Q	**13.22**	**39.2**	**65.7**	72.8	**59.0**	32.4	**71.8**	33.1	**66.1**	**55.01**
ReSC-Q	16.15	33.4	60.6	**74.3**	55.7	**34.4**	69.7	**34.4**	64.1	53.33
Qwen3-14B (gs: 128)										
QuaRot + ReSC-Q	**11.05**	**50.8**	**74.5**	77.2	**66.9**	**40.4**	**74.2**	**39.0**	**70.2**	**61.66**
ReSC-Q	11.89	43.3	60.9	**83.4**	65.4	39.0	73.7	37.5	68.6	58.97

D.8 Compatibility with Rotation-Based Quantization

We examine whether ReSC-Q is compatible with rotation-based quantization. We apply ReSC-Q after QuaRot (15) under 2-bit quantization on Llama-3.1-8B-Instruct and Qwen3-14B, using group sizes 32 and 128. QuaRot is a training-free rotation-based quantization method that applies orthogonal transformations to redistribute outliers and homogenize the coordinate-wise weight and activation distributions. Since these rotations preserve the original full-precision model function, they can be applied as a preprocessing step before weight quantization, producing a representation that is generally easier to quantize. *QuaRot + ReSC-Q* means that we first apply the same rotations used by QuaRot and then quantize the resulting rotated weights using ReSC-Q.

Table 10 shows that ReSC-Q remains highly effective after rotation. At group size 32, applying ReSC-Q after QuaRot substantially improves both perplexity and average zero-shot accuracy over QuaRot alone. However, rotation provides limited additional benefit over ReSC-Q itself. On Llama-3.1-8B, rotation improves perplexity from 11.76 to 11.32 but slightly reduces average zero-shot accuracy from 58.83 to 58.64. On Qwen3-14B, ReSC-Q alone remains marginally better in both metrics. This

suggests that the benefit of rotation is limited when the affine groups are already small enough to reduce within-group heterogeneity.

Rotation becomes more beneficial as the group size increases. At group size 128, QuaRot + ReSC-Q improves perplexity and average zero-shot accuracy from 16.15 and 53.33 to 13.22 and 55.01 on Llama-3.1-8B, and from 11.89 and 58.97 to 11.05 and 61.66 on Qwen3-14B. With a larger group, a single scale and zero-point must accommodate a more heterogeneous set of weights, often including multiple salient coordinates that cannot be represented well by one affine grid. Rotation redistributes weight and activation magnitudes, reducing within-group heterogeneity and making the shared affine quantizer easier to fit. Thus, rotation and ReSC-Q become more complementary as the group size increases.

Table 11: Comparison with structured-code quantization on Llama-3.1-8B-Instruct. All methods target nominal 2-bit weights and use no fine-tuning; ReSC-Q uses group size 32, while QuIP# and QTIP retain their native lattice and trellis representations. Each entry reports PPL↓ / average zero-shot accuracy↑.

Method	**Representation**	$n = 4$	$n = 8$	$n = 128$
ReSC-Q	Uniform affine	13.34 / 56.57	12.89 / 58.52	11.76 / 58.80
QuIP#	E8P lattice	17.96 / 48.15	14.23 / 53.51	11.55 / 57.96
QuIP# + ReSC-Q	E8P lattice + weighted objective	12.59 / 55.40	12.27 / 57.55	11.19 / 59.09
QTIP	Trellis-coded	<u>12.06</u> / <u>55.43</u>	<u>11.01</u> / <u>57.82</u>	**9.82** / <u>61.81</u>
QTIP + ReSC-Q	Trellis-coded + weighted objective	**10.54 / 59.26**	**10.46 / 60.63**	<u>9.85</u> / **61.85**

D.9 Comparison with Structured-Code Quantization

QuIP# (13) and QTIP (14) are structured-code quantization methods designed for extreme low-bit compression. QuIP# quantizes eight-dimensional weight vectors using an E_8-lattice-based codebook, whereas QTIP represents a long sequence of weights using trellis-coded quantization and a stateful decoder. At 2 bits per weight, these structured representations provide a richer feasible code space than the four scalar levels of uniform affine quantization. We use these methods both as structured-code references and as alternative quantization representations for evaluating the residual-aware objective of ReSC-Q.

We examine whether the residual-aware objective of ReSC-Q can be incorporated into these structured quantizers. In the combined variants, we preserve the original incoherence processing and the native lattice or trellis representation, but replace the Euclidean codeword-selection cost with the ReSC-Q-weighted distortion

$$\arg\min_{\mathbf{g}\in\mathcal{C}} \|\mathbf{x}-\mathbf{g}\|_2^2 \quad \longrightarrow \quad \arg\min_{\mathbf{g}\in\mathcal{C}} \sum_v d_v (x_v - g_v)^2. \tag{92}$$

Specifically, d_v is incorporated into the E8P lattice-search cost for QuIP# and into the Viterbi path cost for QTIP. The corresponding cross-group curvature is regularized using the same ReSC-Q shrinkage procedure.

Table 11 shows that the choice of quantization representation and the residual-aware objective of ReSC-Q are complementary. With limited calibration data, transferring the residual-aware weighting to the lattice and trellis objectives with cross-group shrinkage substantially improves both structured quantizers. The gain remains consistent for QuIP# as the calibration budget increases. For QTIP, the weighted variant provides a large improvement at small calibration budgets, while it becomes nearly identical to the original QTIP at $n = 128$, where the trellis quantizer is already highly accurate.

These results show that the residual-aware objective is complementary to the underlying quantization representation. ReSC-Q uses group-wise affine quantization by default, while QuIP# and QTIP provide more expressive lattice and trellis code spaces with representation-specific decoding. The improvements obtained by applying the same weighted objective to these representations demonstrate that the proposed within-group objective is not specific to affine quantization. These representations therefore provide different quality–deployment trade-offs rather than directly interchangeable quantization formats.

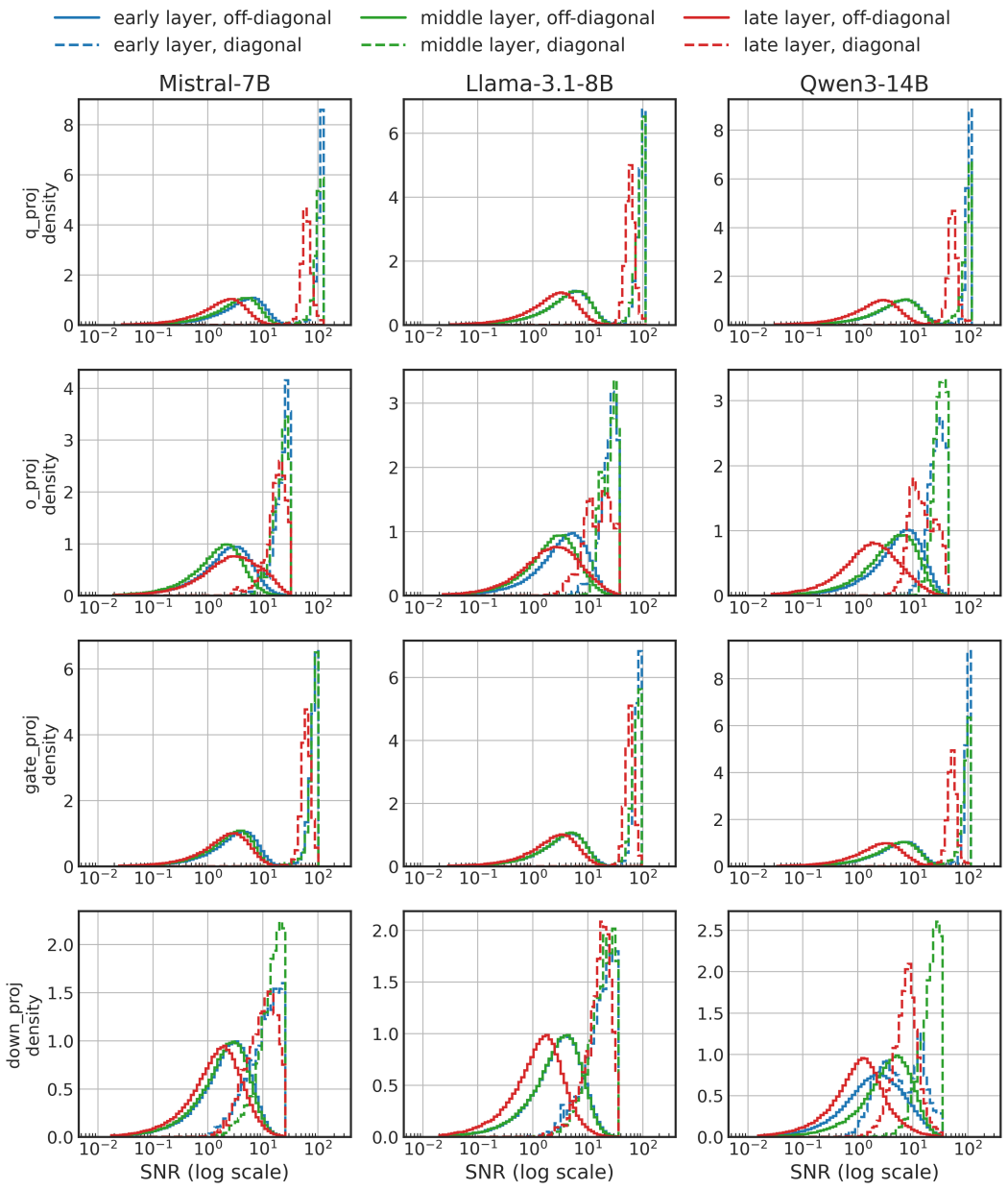

Figure 5: Entry-wise Hessian signal-to-noise ratio (SNR) for diagonal and off-diagonal entries across models, layers, and projection modules. Results are shown for `q_proj`, `o_proj`, `gate_proj`, and `down_proj` at representative early, middle, and late layers of `Mistral-7B`, `Llama-3.1-8B`, and `Qwen3-14B`. Solid and dashed curves denote off-diagonal and diagonal entries, respectively.

D.10 Broader Hessian Reliability Diagnostics

We further examine the reliability of Hessian estimates across different models, layers, and projection modules. Fig. 5 reports the entry-wise SNR distributions for `q_proj`, `o_proj`, `gate_proj`, and `down_proj` at representative early, middle, and late layers of `Mistral-7B`, `Llama-3.1-8B`, and `Qwen3-14B`. Across these settings, off-diagonal entries consistently exhibit lower SNR than diagonal entries, indicating larger sampling variation relative to their mean magnitude. This result

shows that the reliability gap between diagonal and off-diagonal curvature persists across different models and projection modules.

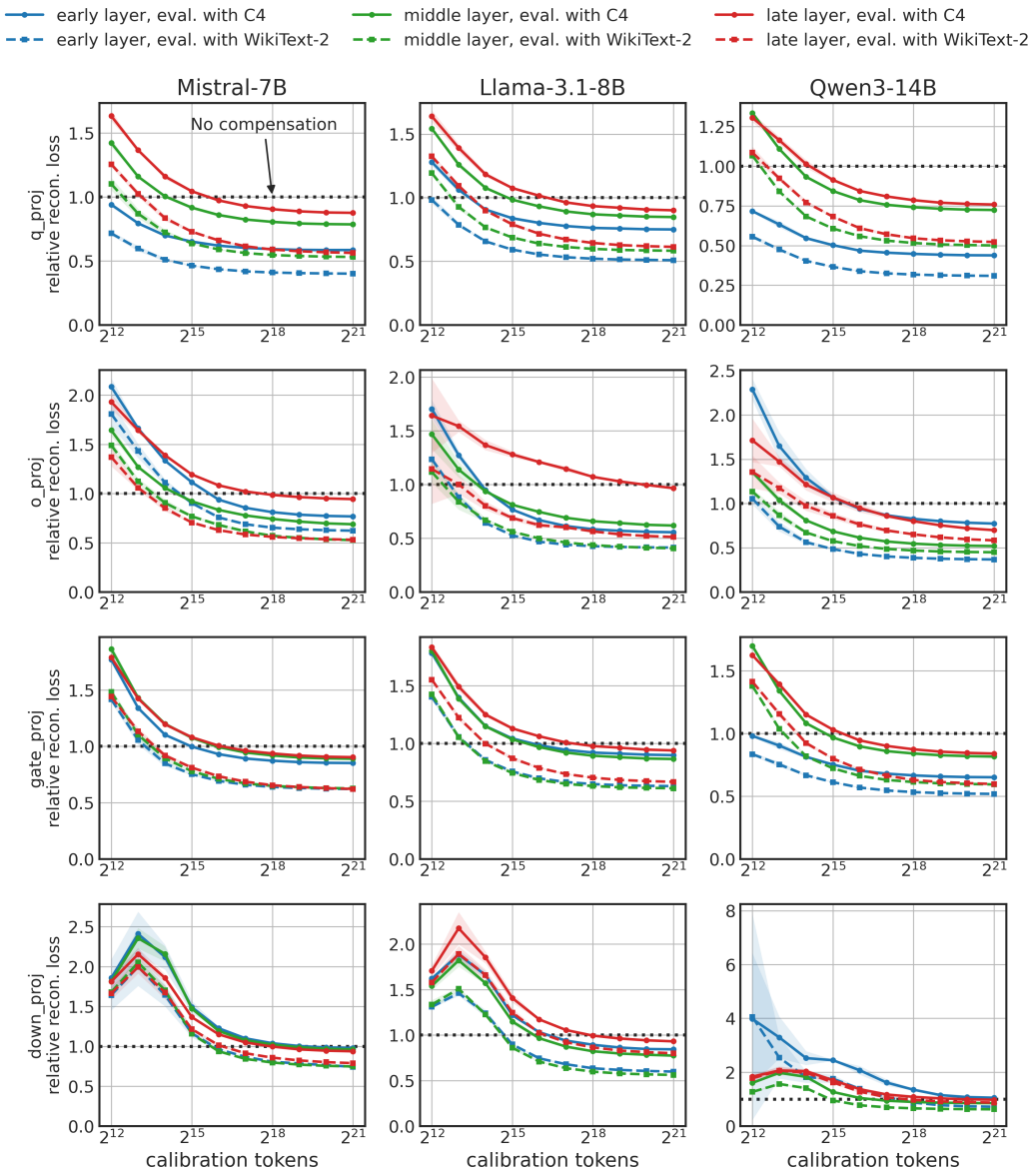

Figure 6: Relative reconstruction loss after unshrunk error compensation with varying calibration sizes. Hessian estimates are computed from WikiText-2, and the resulting compensation is evaluated on both WikiText-2 and C4. Results are shown for `q_proj`, `o_proj`, `gate_proj`, and `down_proj` at representative early, middle, and late layers of `Mistral-7B`, `Llama-3.1-8B`, and `Qwen3-14B`. Solid and dashed curves denote evaluation on C4 and WikiText-2, respectively. The loss is normalized by the corresponding loss without compensation.

D.11 Broader Evaluation of Error Compensation Generalization

We evaluate whether the calibration-dependent behavior of Hessian-based error compensation generalizes across models, projection modules, layers, and evaluation datasets. Figs. 6 and 7 evaluate the same unshrunk compensation procedure using WikiText-2 and C4 as the calibration data, respectively.

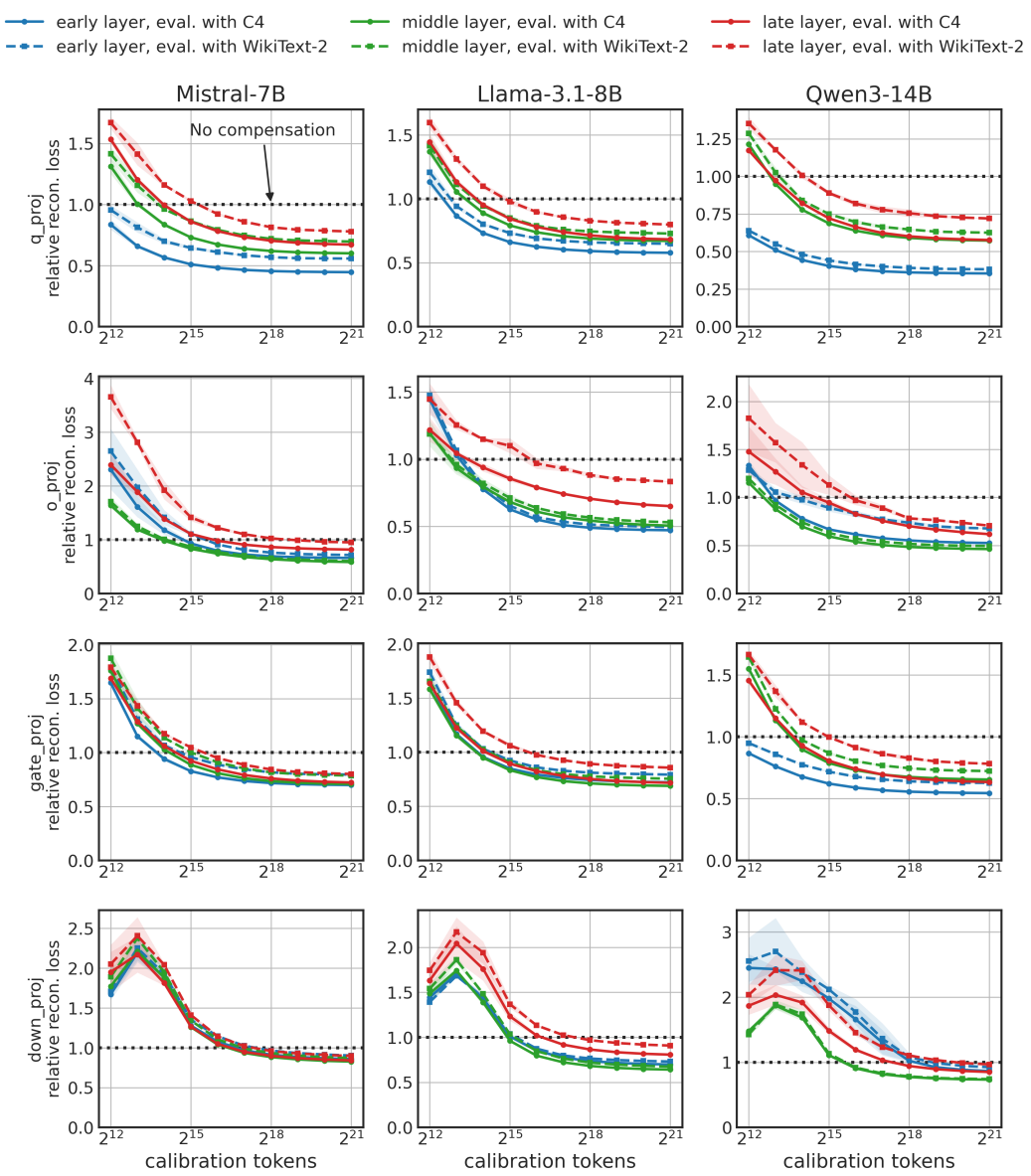

Figure 7: Relative reconstruction loss after unshrunk error compensation with varying calibration sizes. Hessian estimates are computed from C4, and the resulting compensation is evaluated on both WikiText-2 and C4. Results are shown for `q_proj`, `o_proj`, `gate_proj`, and `down_proj` at representative early, middle, and late layers of `Mistral-7B`, `Llama-3.1-8B`, and `Qwen3-14B`. Solid and dashed curves denote evaluation on C4 and WikiText-2, respectively. The loss is normalized by the corresponding loss without compensation.

For each calibration size, we compute the compensation update from the calibration Hessian and evaluate the resulting reconstruction loss on both datasets.

Across the evaluated models and projection modules, error compensation can increase reconstruction loss with limited calibration data. Increasing the calibration size generally reduces this degradation, but the improvement depends on both the model component and the evaluation dataset. In particular, the compensation obtained from one dataset does not always provide the same reduction in reconstruction loss on the other dataset. These results further support the observation in the main text that

minimizing reconstruction loss on the calibration data does not necessarily yield the same behavior on unseen inputs.

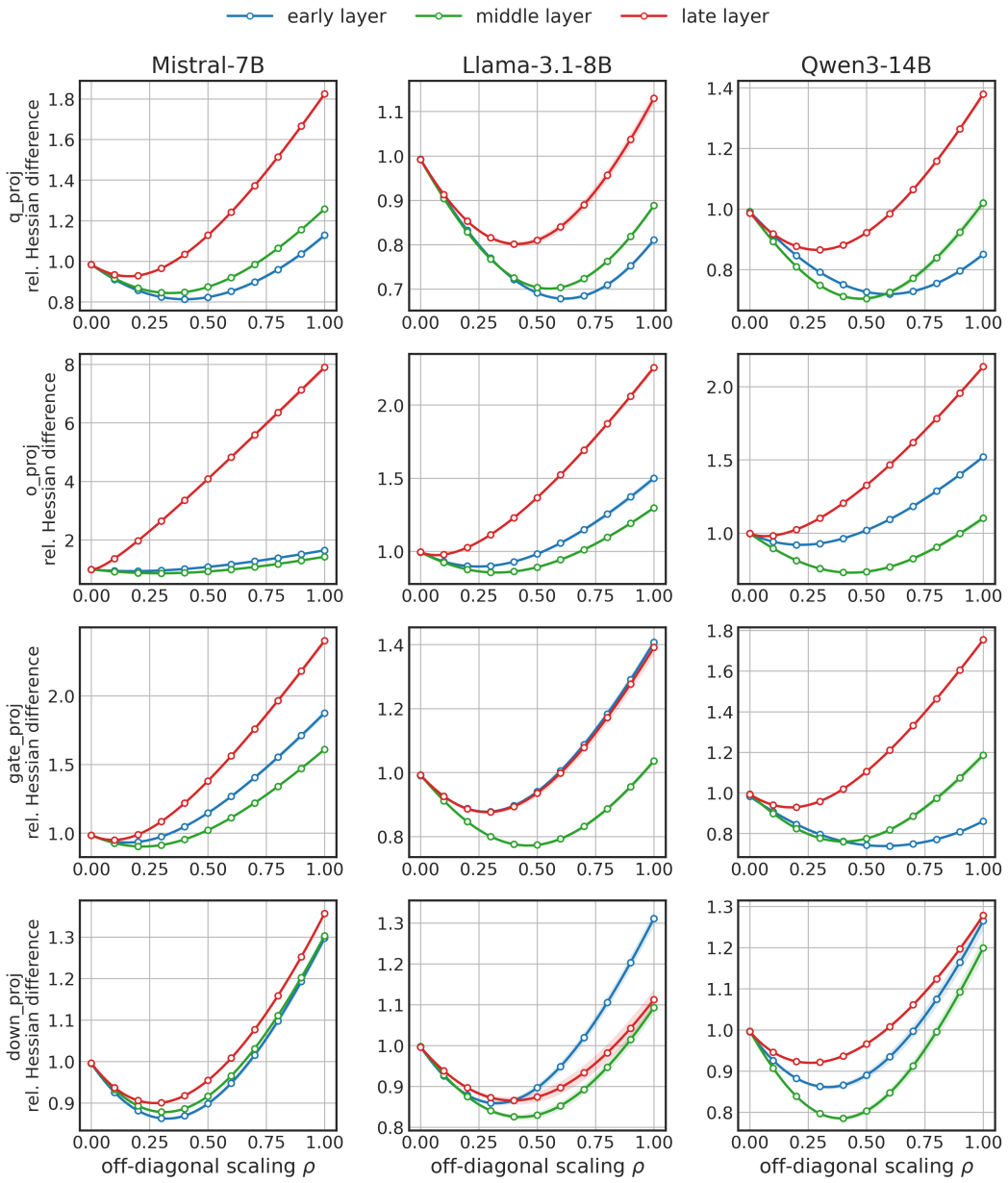

Figure 8: Relative ℓ_1 Hessian difference as a function of the off-diagonal scaling coefficient ρ. The Hessian estimated from 2^{18} WikiText-2 tokens is compared with a C4 reference Hessian computed from 2^{24} tokens. Results are shown for `q_proj`, `o_proj`, `gate_proj`, and `down_proj` at representative early, middle, and late layers of `Mistral-7B`, `Llama-3.1-8B`, and `Qwen3-14B`. The diagonal entries are kept unchanged while the off-diagonal entries are scaled by ρ.

D.12 Shrinkage Behavior Across Layers and Modules

We examine whether the shrinkage behavior observed in Fig. 1(d) generalizes across models, layers, and projection modules. Fig. 8 reports the relative ℓ_1 difference between the shrunk WikiText-2 Hessian estimate and the C4 reference as a function of ρ.

Across the evaluated settings, the Hessian difference is generally minimized at an intermediate value of ρ, rather than at either $\rho = 0$ or $\rho = 1$. The minimizing value varies across layers and projection modules, showing that the useful amount of off-diagonal shrinkage is not universal across network components. This observation is consistent with estimating the shrinkage coefficient from calibration statistics rather than fixing a single global value.