# CoreQ: Mismatch-Corrected Successive Rounding for LLM Quantization

**Seohyeon Cha**[1] **Huancheng Chen**[1] **Dongjun Kim**[1] **Haoran Zhang**[1]
**Kevin Chan**[2] **Gustavo de Veciana**[1] **Haris Vikalo**[1]
[1]The University of Texas at Austin [2]DEVCOM Army Research Laboratory

## Abstract

Post-training quantization (PTQ) enables efficient deployment of large language models by mapping pretrained weights to low-bit formats without retraining, typically using a small calibration set to minimize a layer-wise calibration objective. However, this sequential procedure induces a mismatch: errors from earlier quantized layers alter the inputs received by later layers, causing the activations to deviate from those of the full-precision model. Recent approaches introduce mismatch-aware calibration objectives to compensate for this effect, but leave open how much of the observed mismatch should shift each layer's calibration target. We show that fully applying the empirical mismatch correction can overfit finite calibration data, while using a fixed correction coefficient ignores variation in mismatch estimates across layers. To address these limitations, we propose CoreQ, a learning-free PTQ framework that uses a geometric decomposition of the mismatch to compute a closed-form, layer-adaptive correction coefficient. This coefficient controls how much of the observed mismatch enters each layer's calibration target, reducing overfitting to limited calibration data without coefficient tuning. CoreQ then applies efficient greedy successive rounding to the resulting triangular least-squares objective. Its bounded beam-search extension, K-CoreQ, trades modest additional compute for improved performance. Across multiple LLM families, scales, bit-widths, and quantization settings, CoreQ and K-CoreQ achieve lower perplexity and higher downstream accuracy than strong PTQ baselines. Code is available at CoreQ.

## 1 Introduction

Large language models (LLMs) achieve strong performance across a wide range of tasks, but their size presents a major challenge to practical deployment [1, 2, 3]. Modern transformer-based LLMs contain billions of parameters, requiring tens to hundreds of gigabytes of weight storage when stored in high precision and incurring substantial inference latency due to memory traffic [4, 5]. These costs limit deployment on commodity hardware and make on-device or edge inference difficult [6]. Consequently, reducing memory footprint and runtime while preserving accuracy has become essential for practical LLM deployment.

Post-training quantization (PTQ) enables compression of a pretrained model without retraining by mapping its weights to low-bit formats, providing a practical path to efficient inference. To remain scalable, PTQ is typically performed layer-by-layer. Given a small calibration set, sequential PTQ quantizes each layer by solving a discrete rounding problem that minimizes a layer-wise calibration objective using activations from the already-quantized prefix [7, 8, 9, 10, 11]. While efficient, this layer-wise procedure ignores the compounding effect of earlier quantization errors across layers. As layers are quantized sequentially, the activations entering deeper layers deviate from those in the full-precision model, creating a mismatch between the full-precision and partially quantized computation paths [12, 13, 14].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Recent methods, such as GPTAQ [12] and QEP [13], address this problem by redefining the calibration objective: they use the full-precision layer output on full-precision activations as the calibration target, while evaluating the quantized layer on activations produced by the quantized prefix. This incorporates the upstream-induced mismatch into layer-wise PTQ, but leaves open how strongly the observed mismatch should shift the calibration target under finite calibration data. Applying the full correction can overfit calibration data, while using a fixed correction coefficient cannot adapt to variation across layers and model families. Other approaches improve the layer-wise proxy through richer curvature estimates [15, 16, 17] or by quantizing multiple layers jointly [18], but these strategies either increase the cost of the layer-wise objective or change the quantization unit.

Even with a well-chosen calibration objective, PTQ still requires solving a *discrete* optimization problem over low-bit weight grids. Most methods obtain quantized weights by sequentially rounding the weight matrix, either through greedy column-wise rounding [11, 19, 20, 21] or via iterative refinement schemes such as coordinate descent [22, 23]. The rounding strategy directly affects quantization performance: poor early decisions can propagate across weight columns and substantially degrade the quality of the solution. Thus, improving PTQ requires not only a well-defined calibration objective, but also a more reliable way to search over the low-bit weight space.

To address these limitations, we propose **CoreQ**, a learning-free PTQ framework that adaptively controls how strongly mismatch correction shifts each layer's calibration target. CoreQ computes a closed-form correction coefficient, $\alpha_{\mathrm{corr}}$, by decomposing the mismatch into a component reachable by changing the current layer's weights and an orthogonal residual. This provides a layer-adaptive alternative to full mismatch correction and fixed scaling, without model-specific tuning, gradient updates, or validation search. We then solve the resulting discrete rounding problem using successive-rounding algorithms. While CoreQ provides a greedy solution, **K-CoreQ** extends it with bounded beam search by maintaining multiple partial rounding paths. This expanded search reduces error accumulation from early weight choices and provides a practical trade-off between computational efficiency and quantization quality.

Our contributions are summarized as follows:

- **Finite-calibration analysis.** We show theoretically and empirically that fully compensating for mismatch on finite calibration data leads to overfitting (Section 3.1 and Appendix C.1). The analysis motivates a flexible, data-dependent mismatch correction strategy.
- **Layer-adaptive mismatch correction.** We introduce CoreQ, a learning-free PTQ framework that computes a layer-wise correction coefficient with no need for gradient updates or validation search.
- **Successive rounding.** Given this correction coefficient, CoreQ solves the rounding problem induced by the mismatch-corrected calibration target by exploiting the triangular structure of a least-squares proxy. This yields an efficient greedy successive-rounding solver and a bounded beam-search extension, $K$-CoreQ, which provides a controllable performance-cost trade-off (Section 3.2).
- **Extensive empirical validation.** Across multiple LLM families, model scales, bit-widths, and quantization settings, CoreQ and $K$-CoreQ improve perplexity and downstream task accuracy over strong PTQ baselines. For example, on 2-bit quantized LLaMA-2-70B, we achieve a **48–56%** improvement in WikiText-2 perplexity relative to the best baseline (Section 4).

## 2 Background: Calibration Objectives for Layer-Wise PTQ

We consider sequential, layer-wise post-training quantization (PTQ) of a transformer model. All quantities below refer to a single layer; we suppress the layer index $l$ for clarity. Let $\mathbf{W} \in \mathbb{R}^{m\times n}$ denote the full-precision weight matrix of the current layer, and let $\mathbf{Q} \in \mathcal{Q}$ denote its quantized counterpart on a discrete low-bit grid.

Let $\mathbf{X}_{\mathrm{q}} = [x_{\mathrm{q}}^{(1)}, \ldots, x_{\mathrm{q}}^{(N)}] \in \mathbb{R}^{n\times N}$ denote the matrix of $N$ calibration activation vectors entering the current layer after all preceding layers have been quantized. The layer-wise PTQ problem can be written in the common form $\min_{\mathbf{Q}\in\mathcal{Q}} \|\mathbf{Y} - \mathbf{Q}\mathbf{X}_{\mathrm{q}}\|_F^2$, where $\mathbf{Y}$ is the *calibration target*, i.e., the desired output that the quantized layer is asked to match on the quantized-prefix activation $\mathbf{X}_{\mathrm{q}}$.

**Standard Calibration Objective.** Standard PTQ methods use the full-precision output on the same quantized-prefix activations as the calibration target, i.e., $\mathbf{Y} = \mathbf{W}\mathbf{X}_{\mathrm{q}}$. This gives the standard

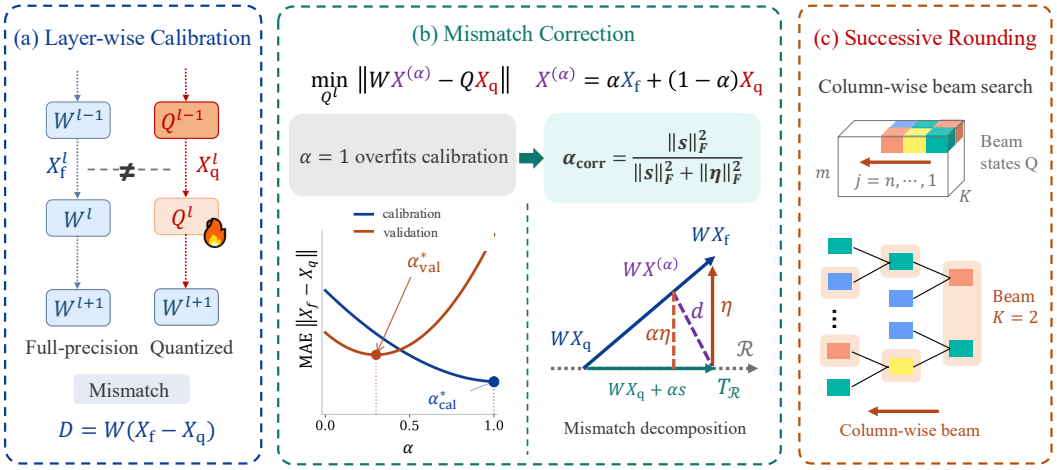



Figure 1: Overview of CoreQ. (a) Sequential layer-wise quantization creates the mismatch between the full-precision and partially quantized activations. (b) CoreQ computes $\alpha_{\text{corr}}$ to scale the mismatch correction according to the component reproducible by changing the current layer's weights. (c) The resulting corrected target induces a triangular least-squares rounding problem, efficiently solved by CoreQ or $K$-CoreQ.

calibration objective [7]

$$\mathcal{L}_{\text{std}}(\mathbf{Q}; \mathbf{X}_{\text{q}}) := \|(\mathbf{W} - \mathbf{Q})\mathbf{X}_{\text{q}}\|_F^2 = \text{tr}\big((\mathbf{W} - \mathbf{Q})\mathbf{H}(\mathbf{W} - \mathbf{Q})^\top\big), \tag{1}$$

where $\mathbf{H} := \mathbf{X}_{\text{q}}\mathbf{X}_{\text{q}}^\top$ serves as the Hessian proxy. Solvers such as GPTQ [11] and LDLQ [19] approximately minimize (1) by sequential rounding guided by this proxy [15, 16].

**Mismatch-Aware Calibration Objective.** Let $\mathbf{X}_{\text{f}} \in \mathbb{R}^{n \times N}$ denote the full-precision activations corresponding to the same calibration tokens as $\mathbf{X}_{\text{q}}$. Since the standard calibration target is built from $\mathbf{X}_{\text{q}}$, it does not include the upstream-induced *mismatch*

$$\mathbf{D} := \mathbf{W}(\mathbf{X}_{\text{f}} - \mathbf{X}_{\text{q}}),$$

which captures the effect of preceding quantized layers at the current layer [18, 14]. Mismatch-aware calibration incorporates this mismatch by using the target $\mathbf{Y} = \mathbf{W}\mathbf{X}_{\text{f}}$. The resulting mismatch-aware calibration objective is [12, 13]

$$\mathcal{L}_{\text{mis}}(\mathbf{Q}; \mathbf{X}_{\text{f}}, \mathbf{X}_{\text{q}}) := \|\mathbf{W}\mathbf{X}_{\text{f}} - \mathbf{Q}\mathbf{X}_{\text{q}}\|_F^2 = \|(\mathbf{W} - \mathbf{Q})\mathbf{X}_{\text{q}} + \mathbf{D}\|_F^2. \tag{2}$$

Thus, both objectives optimize the quantized layer on $\mathbf{X}_{\text{q}}$, but differ in the calibration target: standard calibration uses $\mathbf{Y} = \mathbf{W}\mathbf{X}_{\text{q}}$, while mismatch-aware calibration uses $\mathbf{Y} = \mathbf{W}\mathbf{X}_{\text{f}}$.

## 3 CoreQ: Mismatch-Corrected Calibration and Successive Rounding

Section 2 introduced two endpoint calibration objectives: standard calibration, which uses the target $\mathbf{W}\mathbf{X}_{\text{q}}$, and mismatch-aware calibration, which uses the target $\mathbf{W}\mathbf{X}_{\text{f}}$. We start from the observation that neither endpoint is generally reliable: the former ignores the mismatch, while the latter can overfit finite calibration data. CoreQ addresses this by constructing an interpolated calibration target with a closed-form correction coefficient, avoiding manual tuning or validation search.

### 3.1 Calibration Target Selection

**Orthogonal decomposition of the mismatch.** From (2), the current layer can offset the mismatch $\mathbf{D}$ only through the term $(\mathbf{W} - \mathbf{Q})\mathbf{X}_{\text{q}}$. Since $\mathbf{X}_{\text{q}}$ is fixed during layer-wise calibration, all output changes realizable by changing the current layer's weights lie in the *reachable subspace* $\mathcal{R} := \{\mathbf{A}\mathbf{X}_{\text{q}} : \mathbf{A} \in \mathbb{R}^{m \times n}\}$. The mismatch $\mathbf{D}$ need not lie entirely in this subspace. Projecting it onto $\mathcal{R}$ yields the orthogonal decomposition

$$\mathbf{D} = \widetilde{\mathbf{S}}\mathbf{X}_{\text{q}} + \boldsymbol{\eta}, \quad \widetilde{\mathbf{S}} := \mathbf{D}\mathbf{X}_{\text{q}}^\top(\mathbf{X}_{\text{q}}\mathbf{X}_{\text{q}}^\top)^{-1}, \quad \boldsymbol{\eta} \perp \mathcal{R}, \tag{3}$$

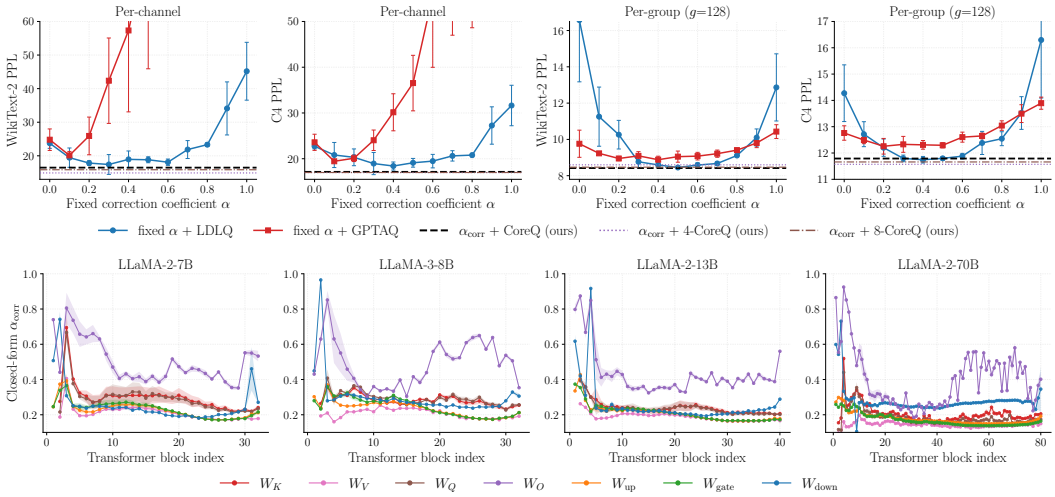



Figure 2: (Top) Fixed-$\alpha$ baselines use globally chosen correction coefficient on LLaMA-3-8B. CoreQ uses a per-layer $\alpha_{\text{corr}}$ and matches or improves upon the best fixed $\alpha$ across granularities and datasets. (Bottom) $\alpha_{\text{corr}}$ varies substantially across layer types and depths, with similar patterns across LLaMA model variants. This supports using layer-dependent correction coefficients. Additional observations are in Appendix C.1.

where $\widetilde{\mathbf{S}}\mathbf{X}_{\text{q}} \in \mathcal{R}$ is the part of the mismatch absorbable by changing the current layer's weights, while $\boldsymbol{\eta}$ is the residual component.

Substituting this decomposition into (2) separates the reachable component from the residual:

$$\mathcal{L}_{\text{mis}}(\mathbf{Q}) = \|(\mathbf{W} + \widetilde{\mathbf{S}})\mathbf{X}_{\text{q}} - \mathbf{Q}\mathbf{X}_{\text{q}}\|_F^2 + \|\boldsymbol{\eta}\|_F^2. \tag{4}$$

Since the residual term is independent of $\mathbf{Q}$, minimizing $\mathcal{L}_{\text{mis}}$ is equivalent to using the *reachable calibration target* $\mathbf{Y}_{\mathcal{R}} := (\mathbf{W} + \widetilde{\mathbf{S}})\mathbf{X}_{\text{q}}$. However, $\widetilde{\mathbf{S}}$ is estimated from finite calibration data, so using $\mathbf{Y}_{\mathcal{R}}$ without shrinkage can overfit finite-calibration error.

**Choosing the correction coefficient.** To control reliance on this finite-calibration estimate, we consider calibration targets along the path $\mathbf{Y}^{(\alpha)} := \mathbf{W}\mathbf{X}^{(\alpha)}$, where $\mathbf{X}^{(\alpha)} := (1-\alpha)\mathbf{X}_{\text{q}} + \alpha\mathbf{X}_{\text{f}}$ for $\alpha \in [0, 1]$. Using $\mathbf{D} = \widetilde{\mathbf{S}}\mathbf{X}_{\text{q}} + \boldsymbol{\eta}$, this target can be written as

$$\mathbf{Y}^{(\alpha)} = \mathbf{W}\mathbf{X}_{\text{q}} + \alpha\mathbf{D} = (\mathbf{W} + \alpha\widetilde{\mathbf{S}})\mathbf{X}_{\text{q}} + \alpha\boldsymbol{\eta}. \tag{5}$$

Here, the *correction coefficient* $\alpha$ controls how far the calibration target moves from standard calibration toward mismatch-aware calibration: $\alpha = 0$ gives (1), while $\alpha = 1$ gives (2). The next theorem formalizes the trade-off between ignoring the mismatch and fully incorporating its finite-calibration estimate, showing that the population-optimal coefficient is generally smaller than 1 under finite-calibration error.

**Theorem 3.1** (Intermediate correction under finite-calibration error)**.** *Let* $\bar{\mathbf{H}} := \mathbb{E}[X_{\text{q}}X_{\text{q}}^\top] \succ 0$ *and* $\bar{\mathbf{S}} := \mathbf{W}\mathbb{E}[(X_{\text{f}} - X_{\text{q}})X_{\text{q}}^\top]\bar{\mathbf{H}}^{-1}$. *The population layer-wise error* $\mathcal{L}_{\text{pop}}(\widehat{\mathbf{W}}) := \mathbb{E}\|\mathbf{W}X_{\text{f}} - \widehat{\mathbf{W}}X_{\text{q}}\|_2^2$ *is minimized over continuous weights by* $\mathbf{W}^\star = \mathbf{W} + \bar{\mathbf{S}}$. *Assume* $\mathbb{E}[\widetilde{\mathbf{S}}] = \bar{\mathbf{S}}$. *For continuous weights of the form* $\mathbf{W} + \alpha\widetilde{\mathbf{S}}$, *the expected excess population error is*

$$R(\alpha) := \mathbb{E}_{\widetilde{\mathbf{S}}}\Big[\mathcal{L}_{\text{pop}}(\mathbf{W} + \alpha\widetilde{\mathbf{S}}) - \mathcal{L}_{\text{pop}}(\mathbf{W}^\star)\Big] = (1-\alpha)^2\mu^2 + \alpha^2\sigma^2.$$

*Therefore, the optimal correction coefficient is* $\alpha^\star := \arg\min_{\alpha\in[0,1]} R(\alpha) = \frac{\mu^2}{\mu^2+\sigma^2}$, *where* $\mu^2 := \|\bar{\mathbf{S}}\bar{\mathbf{H}}^{1/2}\|_F^2$ *is the ideal correction energy and* $\sigma^2 := \mathbb{E}_{\widetilde{\mathbf{S}}}\|(\widetilde{\mathbf{S}} - \bar{\mathbf{S}})\bar{\mathbf{H}}^{1/2}\|_F^2$ *is the finite estimation error energy. Therefore,* $0 < \alpha^\star < 1$ *whenever* $\mu^2 > 0$ *and* $\sigma^2 > 0$.

The proof is provided in Appendix A. Theorem 3.1 is an idealized result: under unbiased correction estimation, it characterizes the trade-off between applying too little correction and amplifying

estimation error from finite calibration data. The optimal coefficient depends on the correction and estimation error energies of each layer, motivating a correction strength that adapts across layers. The theorem assumes an unbiased estimator $\mathbb{E}[\widetilde{\mathbf{S}}] = \bar{\mathbf{S}}$, which does not hold exactly after matrix inversion. Appendix A.3 extends the analysis to biased estimators and shows that the resulting deviation vanishes at rate $O(N^{-1})$ in the calibration size. In either case, however, $\mu^2$ and $\sigma^2$ are population quantities that cannot be evaluated during quantization.

CoreQ therefore uses a criterion motivated by the mismatch decomposition in (3). For each layer, CoreQ chooses $\alpha$ to minimize the squared distance between $\mathbf{Y}^{(\alpha)}$ and the reachable calibration target $\mathbf{Y}_{\mathcal{R}}$, as illustrated in Figure 1(b). Writing $\mathbf{s} := \widetilde{\mathbf{S}}\mathbf{X}_{\mathrm{q}}$, the squared distance is

$$\|\mathbf{d}(\alpha)\|_F^2 = (1-\alpha)^2\|\mathbf{s}\|_F^2 + \alpha^2\|\boldsymbol{\eta}\|_F^2. \tag{6}$$

This criterion selects the correction coefficient by balancing two effects: the first term penalizes leaving the reachable component of the mismatch uncorrected, while the second penalizes the residual component, which cannot be corrected by changing the current layer's weights. Minimizing this distance gives the closed-form correction coefficient

$$\alpha_{\mathrm{corr}} = \frac{\|\mathbf{s}\|_F^2}{\|\mathbf{s}\|_F^2 + \|\boldsymbol{\eta}\|_F^2} = \frac{\|\widetilde{\mathbf{S}}\mathbf{X}_{\mathrm{q}}\|_F^2}{\|\widetilde{\mathbf{S}}\mathbf{X}_{\mathrm{q}}\|_F^2 + \|\boldsymbol{\eta}\|_F^2}. \tag{7}$$

In this form, $\|\mathbf{s}\|_F^2$ plays the role of the correction-benefit term in Theorem 3.1, while $\|\boldsymbol{\eta}\|_F^2$ provides a geometric reliability penalty in place of the unobservable estimation error. Thus, $\alpha_{\mathrm{corr}}$ is a computable analogue of the correction–reliability trade-off that can be evaluated from calibration statistics alone (Lemma A.2). Figure 2 and Appendix C.1 show that $\alpha_{\mathrm{corr}}$ varies systematically across layers, reducing the correction when the residual component is large.

### 3.2 Successive Rounding with the Corrected Target

With $\alpha_{\mathrm{corr}}$ selected, CoreQ uses the mismatch-corrected calibration target $\mathbf{Y}^{(\alpha_{\mathrm{corr}})}$ to choose the quantized weights $\mathbf{Q}$ over the low-bit grid. With $\mathbf{W}_{\mathrm{corr}} := \mathbf{W} + \alpha_{\mathrm{corr}}\widetilde{\mathbf{S}}$, optimizing $\mathbf{Q}$ with the corrected target is equivalent to the Hessian-weighted rounding problem

$$\arg\min_{\mathbf{Q}\in\mathcal{Q}} \|\mathbf{Y}^{(\alpha_{\mathrm{corr}})} - \mathbf{Q}\mathbf{X}_{\mathrm{q}}\|_F^2 = \arg\min_{\mathbf{Q}\in\mathcal{Q}} \big\|(\mathbf{W}_{\mathrm{corr}} - \mathbf{Q})\mathbf{L}\big\|_F^2, \quad \mathbf{H} := \mathbf{X}_{\mathrm{q}}\mathbf{X}_{\mathrm{q}}^\top = \mathbf{L}\mathbf{L}^\top. \tag{8}$$

where $\mathbf{L}$ is the lower-triangular matrix obtained from Cholesky decomposition of the Hessian.

**Triangular discrete least squares.** To solve the problem in (8), we exploit its triangular least-squares form. Let $\mathbf{R} := \mathbf{L}^\top$. For row $i$, define the transformed target $\mathbf{y}_i := \mathbf{R}\,\mathbf{W}_{\mathrm{corr},i,:}^\top$ and the quantized row vector $\mathbf{q}_i := \mathbf{Q}_{i,:}^\top$. The Frobenius objective decomposes across rows as $\sum_{i=1}^m \|\mathbf{R}\mathbf{q}_i - \mathbf{y}_i\|_2^2$. Since $\mathbf{R}$ is upper triangular, each row can be rounded sequentially in reverse order. Suppressing the row index $i$, given a fixed suffix $q_{j+1:n}$, define the suffix-conditioned center $c_j$ and incremental cost $\Delta_j$ as

$$c_j(q_{j+1:n}) := \frac{y_j - \sum_{k=j+1}^n \mathbf{R}_{j,k}\, q_k}{\mathbf{R}_{j,j}}, \qquad \Delta_j(q_j \mid q_{j+1:n}) := \mathbf{R}_{j,j}^2\,|q_j - c_j(q_{j+1:n})|^2. \tag{9}$$

The row-wise objective then decomposes as $\|\mathbf{R}\mathbf{q} - \mathbf{y}\|_2^2 = \sum_{j=1}^n \Delta_j(q_j \mid q_{j+1:n})$.

**Greedy decoding.** We approximately solve this discrete optimization problem using greedy successive rounding. At step $j$, given the already rounded suffix $q_{j+1:n}$, **CoreQ** rounds the conditioned center $c_j$ to the nearest available quantization level in the grid $\mathcal{Q}_j$, i.e., it finds

$$q_j^\star(q_{j+1:n}) \in \arg\min_{q\in\mathcal{Q}_j} |q - c_j(q_{j+1:n})|^2. \tag{10}$$

As each weight is quantized, its quantization error shifts the centers of the remaining coordinates yet to be rounded. CoreQ recovers LDLQ when $\alpha_{\mathrm{corr}} = 0$, while $K$-CoreQ further exploits the same triangular structure via bounded beam search in the following.

Table 1: 3-bit symmetric WOQ for LLaMA-2/3 models. We report Wiki2/C4 perplexity (↓), average zero-shot accuracy (↑), and quantization time under per-channel and per-group ($g = 128$) granularity, averaged over 5 random seeds. Bold entries are the best result and values within one standard deviation of the best in each setting. GuidedQ is omitted for per-group setting following the official implementation.

| Model | Method | Per-channel | | | | Per-group (g128) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Wiki2 (↓) | C4 (↓) | Avg.Acc (↑) | Time (s) | Wiki2 (↓) | C4 (↓) | Avg.Acc (↑) | Time (s) |
| L3-8B | FP16 | 6.14 | 8.93 | 74.31 | | 6.14 | 8.93 | 74.31 | |
| | GPTQ | 24.80 ± 3.20 | 23.59 ± 1.77 | 51.68 ± 1.21 | 550.1 | 9.87 ± 0.28 | 12.74 ± 0.40 | 66.82 ± 0.65 | 547.5 |
| | GPTAQ | 33.47 ± 12.40 | 22.00 ± 3.10 | 51.61 ± 1.80 | 701.4 | 9.07 ± 0.33 | 12.15 ± 0.13 | 65.77 ± 1.28 | 715.0 |
| | LDLQ | 24.65 ± 4.79 | 23.16 ± 1.92 | 51.02 ± 1.69 | 457.4 | 14.99 ± 2.85 | 14.12 ± 1.47 | 65.20 ± 2.31 | 472.5 |
| | GuidedQ | 49.79 ± 10.56 | 40.03 ± 6.55 | 48.58 ± 0.96 | 2110.1 | — | — | — | — |
| | **CoreQ** | 16.48 ± 2.04 | **17.13** ± 0.35 | 53.99 ± 1.15 | 564.1 | **8.42** ± 0.16 | 11.79 ± 0.19 | **68.92** ± 0.58 | 575.2 |
| | **4-CoreQ** | **14.90** ± 0.28 | **16.98** ± 0.25 | **55.74** ± 1.01 | 821.5 | 8.60 ± 0.51 | **11.57** ± 0.10 | 68.11 ± 1.65 | 784.0 |
| | **8-CoreQ** | 15.85 ± 1.29 | **16.99** ± 0.39 | **55.24** ± 2.54 | 831.6 | **8.47** ± 0.35 | **11.66** ± 0.25 | **68.70** ± 1.46 | 855.1 |
| L2-7B | FP16 | 5.47 | 6.97 | 70.47 | | 5.47 | 6.97 | 70.47 | |
| | GPTQ | 9.36 ± 0.24 | 10.71 ± 0.09 | 59.41 ± 0.78 | 461.5 | 6.75 ± 0.06 | 8.26 ± 0.01 | 66.14 ± 0.34 | 467.9 |
| | GPTAQ | 8.84 ± 0.23 | 10.04 ± 0.06 | 60.91 ± 0.52 | 578.4 | 6.57 ± 0.03 | 8.04 ± 0.01 | 66.34 ± 0.51 | 590.4 |
| | LDLQ | 9.42 ± 0.17 | 10.65 ± 0.11 | 60.08 ± 0.54 | 401.8 | 6.64 ± 0.10 | 8.12 ± 0.05 | 66.50 ± 0.57 | 427.8 |
| | GuidedQ | 9.01 ± 0.25 | 10.26 ± 0.07 | **61.83** ± 0.48 | 1883.0 | — | — | — | — |
| | **CoreQ** | 8.41 ± 0.21 | 9.61 ± 0.12 | 61.46 ± 0.50 | 467.8 | 6.39 ± 0.03 | 7.86 ± 0.02 | **67.16** ± 0.34 | 470.7 |
| | **4-CoreQ** | **8.03** ± 0.14 | 9.41 ± 0.05 | **62.64** ± 0.89 | 667.6 | **6.31** ± 0.02 | **7.82** ± 0.01 | **67.12** ± 0.24 | 701.2 |
| | **8-CoreQ** | **7.97** ± 0.07 | **9.37** ± 0.01 | **62.53** ± 0.52 | 684.6 | **6.31** ± 0.02 | **7.83** ± 0.02 | **67.05** ± 0.36 | 709.4 |
| L2-13B | FP16 | 4.88 | 6.47 | 73.23 | | 4.88 | 6.47 | 73.23 | |
| | GPTQ | 6.90 ± 0.04 | 8.36 ± 0.02 | 66.26 ± 0.27 | 763.7 | 5.60 ± 0.03 | 7.16 ± 0.01 | **70.93** ± 0.65 | 820.5 |
| | GPTAQ | 6.70 ± 0.04 | 8.20 ± 0.02 | 66.83 ± 0.35 | 1015.8 | 5.54 ± 0.01 | 7.10 ± 0.00 | **71.00** ± 0.11 | 1012.0 |
| | LDLQ | 6.87 ± 0.05 | 8.32 ± 0.02 | 66.97 ± 0.43 | 704.2 | 5.57 ± 0.02 | 7.12 ± 0.00 | **70.97** ± 0.36 | 719.2 |
| | GuidedQ | 6.94 ± 0.02 | 8.37 ± 0.02 | 67.18 ± 0.26 | 3170.4 | — | — | — | — |
| | **CoreQ** | 6.78 ± 0.29 | 8.05 ± 0.10 | 67.03 ± 0.85 | 840.1 | 5.53 ± 0.04 | 7.06 ± 0.03 | **71.19** ± 0.56 | 824.3 |
| | **4-CoreQ** | 6.72 ± 0.18 | **8.01** ± 0.10 | 67.15 ± 0.99 | 1144.0 | **5.51** ± 0.03 | **7.04** ± 0.01 | **70.87** ± 0.27 | 1189.5 |
| | **8-CoreQ** | **6.53** ± 0.07 | **7.98** ± 0.05 | **67.86** ± 0.58 | 1199.2 | **5.50** ± 0.02 | **7.04** ± 0.01 | **71.03** ± 0.30 | 1242.0 |

**Bounded tree search: $K$-CoreQ.** While efficient, greedy successive rounding may suffer from propagation of rounding errors because early errors affect later conditioned centers through the off-diagonal entries of $\mathbf{R}$ [24]. To reduce this effect, we propose **$K$-CoreQ**, which replaces the single greedy path with a bounded beam search that keeps multiple rounding paths.

We view the rounding of each row as a tree search of depth $n$. At coordinate $j$, each beam state stores the already chosen suffix $q_{j+1:n}$ and its accumulated cost. To process coordinate $j$, $K$-CoreQ expands the $K$ retained states over all candidates $q_j \in \mathcal{Q}_j$, scoring each extension according to:

$$S_{j+1}(q_{j+1:n}) := \sum_{t=j+1}^{n} \Delta_t(q_t \mid q_{t+1:n}), \quad S_j(q_{j:n}) = S_{j+1} + \Delta_j(q_j \mid q_{j+1:n}). \tag{11}$$

A `top-k` operation then retains the $K$ lowest-cost partial paths to serve as the beam states for coordinate $j-1$. Keeping multiple partial paths allows $K$-CoreQ to postpone making irreversible choices and reduce the effect of early rounding errors. Since the objective decomposes exactly across rows, the beam states can be tensorized, allowing the level-wise expansion and `top-k` pruning steps to run on GPU across rows. Details of the batched implementation are in Appendix B.2.

# 4 Results

We first evaluate CoreQ in the main PTQ settings in Section 4.2 across model families, scales, bit-widths, and quantization granularities. We then evaluate $K$-CoreQ in Section 4.3, showing how bounded beam search improves rounding quality while controlling runtime and memory. Additional results are provided in Appendix C.

## 4.1 Setup

We evaluate **CoreQ** for post-training quantization of decoder-only LLMs, including LLaMA 2/3 [25, 26], Qwen3-8B [27], and Phi-3 Mini (3.8B) [28]. Calibration uses 128 C4 samples with sequence length 2048. We report perplexity on WikiText2 and C4, and zero-shot accuracy on six commonsense reasoning tasks: WinoGrande [29], PIQA [30], HellaSwag [31], BoolQ [32], ARC-Easy, and ARC-Challenge [33]. We also evaluate CoreQ on the instruction-tuned LLaMA-3.1-8B-Instruct [26] using MMLU [34] (5-shot) and GSM8K [35] with chain-of-thought prompting (8-shot); these results are reported in Table 14 in the Appendix. Results are averaged over 5 seeds. Quantization time is reported as end-to-end wall-clock runtime on a single H200 GPU, including non-GPU overhead. We compare

Table 2: 3-bit per-channel WOQ on Qwen3-8B and Phi3-3.8B. Extended results are provided in Table 12.

| Model | Method | Wiki2 (↓) | C4 (↓) | Q.Time (s) |
|---|---|---|---|---|
| Qwen3-8B | FP16 | 9.73 | 14.65 | – |
| | GPTQ | 18.75 ± 0.45 | 19.69 ± 0.12 | 499.2 |
| | GPTAQ | 17.63 ± 0.59 | 19.08 ± 0.28 | 647.2 |
| | LDLQ | 16.18 ± 0.18 | 18.49 ± 0.08 | 375.4 |
| | **CoreQ** | **15.44** ± 0.23 | **17.51** ± 0.08 | 553.9 |
| Phi3-3.8B | FP16 | 6.01 | 9.11 | – |
| | GPTQ | 15.46 ± 0.50 | 15.80 ± 0.05 | 266.5 |
| | GPTAQ | 12.99 ± 0.28 | 14.21 ± 0.15 | 335.7 |
| | LDLQ | 15.75 ± 0.74 | 16.03 ± 0.48 | 216.1 |
| | **CoreQ** | **11.86** ± 0.34 | **13.68** ± 0.32 | 302.1 |

Table 3: 2/3-bit per-channel WOQ on L2-70B. Extended results are provided in Table 13.

| Bits | Method | Wiki2 (↓) | C4 (↓) | Q.Time (s) |
|---|---|---|---|---|
| 16 | FP16 | 3.32 | 5.52 | – |
| 2 | GPTQ | 181.31 ± 19.81 | 83.17 ± 5.14 | 3301.2 |
| | GPTAQ | 251.82 ± 80.72 | 83.59 ± 11.38 | 4663.4 |
| | LDLQ | 193.51 ± 36.21 | 91.32 ± 10.08 | 2753.5 |
| | **CoreQ** | 94.97 ± 12.49 | **38.46** ± 2.37 | 4731.9 |
| | **4-CoreQ** | **91.68** ± 11.59 | **39.20** ± 4.62 | 6340.9 |
| | **8-CoreQ** | **79.07** ± 14.91 | **37.07** ± 2.54 | 7973.7 |
| 3 | GPTQ | 5.11 ± 0.02 | 6.77 ± 0.02 | 3256.0 |
| | GPTAQ | 5.09 ± 0.05 | 6.73 ± 0.01 | 4727.5 |
| | LDLQ | 5.06 ± 0.02 | 6.73 ± 0.02 | 2855.0 |
| | **CoreQ** | **4.83** ± 0.03 | 6.52 ± 0.01 | 4785.1 |
| | **4-CoreQ** | **4.83** ± 0.02 | 6.52 ± 0.01 | 6264.6 |
| | **8-CoreQ** | **4.82** ± 0.01 | **6.51** ± 0.00 | 8020.5 |

Table 4: Weight–activation–KV-cache quantization. We report Wiki2 perplexity and quantization time for W4A4KV4 and W4A4KV16. Best results and values within one standard deviation of the best are highlighted. OmniQuant is omitted for L3-8B due to perplexity above 100.

| Methods | **W4A4KV4** | | | | | | **W4A4KV16** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | L3-8B | | L2-7B | | L2-13B | | L3-8B | | L2-7B | | L2-13B | |
| | Wiki2 | T (s) | Wiki2 | T (s) | Wiki2 | T (s) | Wiki2 | T (s) | Wiki2 | T (s) | Wiki2 | T (s) |
| OmniQuant | – | – | 15.12 ± 0.21 | 3848 | 11.82 ± 0.06 | 6654 | – | – | 12.21 ± 0.09 | 3640 | 9.95 ± 0.12 | 6337 |
| QuaRot+GPTQ | 8.83 ± 0.19 | 811 | 6.21 ± 0.02 | 1445 | 5.48 ± 0.02 | 2131 | 7.86 ± 0.06 | 793 | 6.08 ± 0.02 | 1317 | 5.35 ± 0.01 | 1977 |
| QuaRot+GPTAQ | 8.08 ± 0.02 | 954 | 5.97 ± 0.00 | 710 | 5.30 ± 0.01 | 1725 | 7.37 ± 0.02 | 918 | 5.87 ± 0.01 | 720 | 5.17 ± 0.01 | 2157 |
| **QuaRot+CoreQ** | **7.96** ± 0.02 | 892 | **5.93** ± 0.01 | 1420 | **5.26** ± 0.00 | 2050 | **7.20** ± 0.01 | 885 | **5.82** ± 0.00 | 1366 | **5.14** ± 0.00 | 2068 |
| SpinQuant+GPTQ | 7.37 ± 0.01 | 577 | 5.95 ± 0.01 | 521 | 5.24 ± 0.01 | 878 | 7.28 ± 0.01 | 573 | 5.89 ± 0.01 | 523 | 5.21 ± 0.01 | 829 |
| SpinQuant+GPTAQ | 7.30 ± 0.00 | 844 | 5.90 ± 0.01 | 753 | 5.22 ± 0.00 | 1121 | 7.20 ± 0.01 | 800 | 5.85 ± 0.00 | 710 | 5.19 ± 0.01 | 1123 |
| **SpinQuant+CoreQ** | **7.26** ± 0.01 | 683 | **5.88** ± 0.00 | 574 | **5.21** ± 0.00 | 1151 | **7.16** ± 0.01 | 690 | **5.82** ± 0.01 | 576 | **5.15** ± 0.00 | 995 |

mainly against learning-free PTQ baselines: GPTQ [11], GPTAQ [12], and LDLQ [19]. Depending on the setting, we also include AWQ [36], OmniQuant [37], and GuidedQuant [15].

### 4.2 Performance Evaluation

**Weight-only quantization.** Table 1 reports 3-bit weight-only quantization (WOQ) under per-channel and per-group ($g$=128) granularity, and the corresponding wall-clock quantization time. On LLaMA-3-8B with per-channel quantization, CoreQ reduces Wiki2 perplexity from 24.65 for the best baseline (LDLQ) to 16.48, **a 33% reduction**; 4-CoreQ further reduces it to 14.90. The results show two trends. First, the benefits of CoreQ grow with the strictness of the quantization constraint. With per-group quantization, where scales are updated every 128 weights, all methods remain stable but the CoreQ variants still achieve the lowest perplexity. With per-channel quantization, GPTQ and LDLQ both exceed 24 perplexity on LLaMA-3-8B, whereas CoreQ and its beam-search variants remain substantially lower. Second, bounded beam-search variants (4-CoreQ, 8-CoreQ) improve over greedy CoreQ, especially in the per-channel setting, while adding moderate runtime overhead and remaining faster than gradient-based alternatives such as GuidedQuant.

**Evaluation on different architectures and scales.** Beyond medium-scale LLaMA models, Tables 2 and 3 show that CoreQ remains effective on Qwen3-8B, Phi-3.8B, and LLaMA-2-70B. On Qwen3-8B with per-channel quantization, CoreQ reduces Wiki2 perplexity by **12.4%** relative to GPTAQ and achieves the lowest C4 perplexity among the compared methods. The same trend holds for Phi-3.8B, where CoreQ achieves the lowest Wiki2 and C4 perplexities, including an **8.7%** Wiki2 reduction over GPTAQ. The benefits are most evident in the extreme low-bit regime. On 2-bit per-channel LLaMA-2-70B, where baseline Wiki2 perplexities exceed 180, CoreQ and $K$-CoreQ reduce Wiki2 perplexity by **48–56%** relative to the best baseline. At 3 bits, CoreQ also achieves the lowest perplexity on both datasets. Overall, the gains are not limited to a single architecture family or model scale, and are especially pronounced in low-bit settings.

**Weight-activation quantization.** We also evaluate the use of CoreQ for weight refinement in weight–activation–KV cache quantization pipelines. We consider two rotation-based preprocessors: QuaRot [38], which applies fixed Hadamard rotations, and SpinQuant [39], which learns rotation matrices to further reduce outliers. For this experiment, calibration uses 128 WikiText2 samples. We apply per-channel symmetric quantization to weights and per-group asymmetric quantization to activations and KV caches. As shown in Table 4, CoreQ improves perplexity over learning-free baselines under both preprocessors at comparable runtimes. OmniQuant, which learns quantization

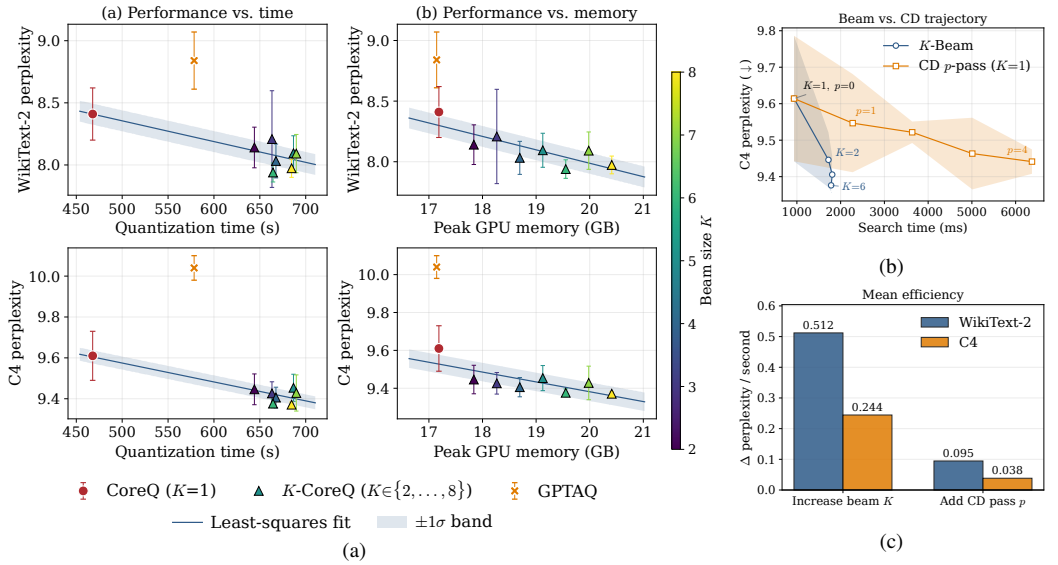



Figure 3: (a) Performance–efficiency trade-offs of $K$-CoreQ for 3-bit L2-7B. WikiText-2 and C4 perplexity are plotted vs. quantization time (left column) and peak GPU memory (right column) as beam width $K$ increases. (b) C4 perplexity vs. search time for increasing beam width $K$ and additional coordinate-descent (CD) passes ($\pm 1$ std. shaded). (c) Mean marginal perplexity improvement per second: beam expansion vs. CD passes.

parameters, requires longer quantization time and achieves higher perplexity in this setting. These results show that CoreQ remains effective in pipelines that also quantize activations and KV caches. However, its gains over the baselines are more modest than in the weight-only experiments. This is consistent with a geometric limit: current-layer weight updates can correct only mismatches in their reachable response space, whereas activation and KV-cache quantization can leave a larger residual outside it (see Appendix C.5). Additional outlier-aware weight-only results compared to AWQ and OmniQuant are provided in Table 10.

**Runtime overhead.** Most of CoreQ's additional calibration cost comes from collecting $\mathbf{X}_{\mathrm{f}}$ and forming the cross-statistic $\mathbf{X}_{\mathrm{f}}\mathbf{X}_{\mathrm{q}}^{\top}$. These mismatch-statistic computations are also required by mismatch-aware baselines such as GPTAQ. In comparison, evaluating $\alpha_{\mathrm{corr}}$ itself accounts for only $1.1\%$ of the profiled GPU time. Table 7 provides a five-stage CUDA-event profile, complementing the end-to-end wall-clock times reported in the main tables.

### 4.3 Performance-cost trade-off of $K$-CoreQ

Here we evaluate $K$-CoreQ, the bounded beam search extension of CoreQ. $K$-CoreQ maintains multiple partial rounding assignments, adding controlled compute and memory to improve the triangular least-squares rounding objective. Figure 3a illustrates this trade-off for 3-bit quantization by plotting perplexity against wall-clock time and peak GPU memory. Increasing $K$ monotonically improves the triangular rounding objective relative to greedy CoreQ in Figure 4 (right). In our experiments, this tighter optimization translates to lower perplexity. Larger beams improve perplexity with modest search-time overhead under batching and increased peak memory due to the expanded beam state. Overall, beam search closes a substantial fraction of the remaining quality gap, suggesting that limited lookahead can reduce errors from early greedy decisions.

**Comparison of rounding algorithms.** To isolate the effect of the discrete search procedure, we compare $K$-CoreQ with cyclic coordinate descent (CD) refinement [22, 19], a common PTQ post-processing step. Figure 3b plots perplexity versus search time as we increase either the beam width $K$ or the number of CD passes $p$. Increasing $K$ results in larger perplexity reductions with small runtime overhead, whereas additional CD passes increase search time to achieve smaller gains. Figure 3c summarizes this trend by reporting the mean marginal perplexity reduction per second for each incremental update. The results indicate that, in the considered settings, bounded beam search uses the search budget more effectively than cyclic coordinate refinement.

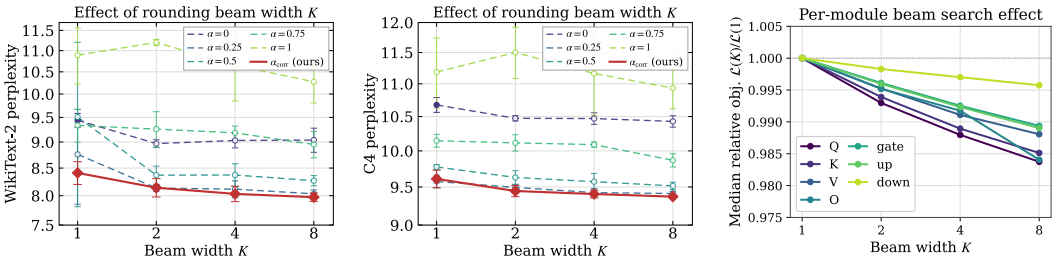



Figure 4: Effect of beam width $K$ on quantization quality for 3-bit per-channel LLaMA-2-7B. **Left, middle:** WikiText-2 and C4 perplexity across correction strengths $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ and $\alpha_{\text{corr}}$. $\alpha_{\text{corr}}$ outperforms every fixed $\alpha$ at every $K$, while increasing $K$ reduces perplexity. **Right:** reduction of the triangular rounding objective under $\alpha = 0$.

**Ablation of calibration and rounding mechanisms.** We isolate the two components of CoreQ by jointly sweeping the per-layer correction coefficient $\alpha \in \{0, 0.25, 0.5, 0.75, 1, \alpha_{\text{corr}}\}$ and the beam width $K \in \{1, 2, 4, 8\}$ for 3-bit per-channel quantization of LLaMA-2-7B (Figure 4, left and center). These two axes correspond to the two decision stages in layer-wise PTQ: choosing the continuous calibration target and rounding the weights to the discrete grid. The ablation shows that both components matter. For a fixed $\alpha$, increasing $K$ generally reduces perplexity relative to greedy rounding, apart from the more variable results at $\alpha = 1$. For a fixed $K$, $\alpha_{\text{corr}}$ yields lower perplexity than the tested constant coefficients, indicating that the closed-form coefficient selects a better calibration target than a uniform correction strength. The best configuration combines $\alpha_{\text{corr}}$ with $K$=8, showing that adaptive calibration and bounded search provide complementary performance improvements. A complementary ablation using GPTQ-based rounding is provided in the Appendix (Table 6).

# 5 Related Work

## 5.1 Layer-wise Calibration Objectives for PTQ

A widely used family of PTQ methods minimizes a Hessian-based calibration objective, often called a layer-wise reconstruction error [7, 8, 9, 10, 11]. GPTQ is the standard LLM-scale realization of this approach. Several extensions incorporate end-to-end information through end-loss gradients, KL-based Hessian approximations, or output cross-entropy distortion [15, 16, 40], but at substantially higher computational cost. More recent methods instead address upstream quantization errors by redefining the layer-wise calibration objective. GPTAQ [12] and Qronos [41, 42] target the observed mismatch by matching the full-precision layer output on full-precision activations to the quantized layer output on activations generated by the quantized prefix. Similarly, QEP [13] introduces a manually tuned coefficient that interpolates between standard Hessian-based objective and full mismatch-aware calibration. Other methods address error propagation via learning-based adapters [14] or joint block quantization [18]. Unlike methods that apply full correction ($\alpha = 1$) or use a fixed global $\alpha$, CoreQ computes a closed-form per-layer correction coefficient $\alpha_{\text{corr}}$ from calibration statistics alone. Learning-based methods such as OmniQuant and LRQ optimize auxiliary parameters by gradient descent [37, 43], trading additional compute for accuracy. An extended discussion is provided in Appendix D.

## 5.2 Discrete Rounding and Search Algorithms

Given a Hessian-based quadratic objective, PTQ reduces to discrete optimization over low-bit weights. GPTQ and its equivalent LDLQ realization are widely used scalar, column-wise greedy rounding solvers [11, 19]. Subsequent work improves rounding quality in two main ways: iterative refinement with coordinate-descent passes which trade additional objective evaluations for lower quantization error [22, 23], and refined error compensation which accounts for first-order effects of quantization error during column-wise updates [21]. A complementary geometric view interprets GPTQ as Babai's nearest-plane method for the closest vector problem, motivating lookahead-based search beyond the greedy path [20, 44]. A separate family of vector quantization methods, including AQLM, QuIP#, and GPTVQ, shifts the discrete problem to codebook decoding and applies beam search over codeword

assignments [45, 46, 47]. Our beam-search extension, $K$-CoreQ, instead performs search over the successive column-wise rounding decisions induced by the triangular least-squares proxy. Rather than committing to one greedy path, as done by Babai's algorithm, $K$-CoreQ maintains $K$ candidate paths through a bounded tree search, leading to a better accuracy-runtime trade-off than coordinate-descent refinement within the same Hessian-based objective.

## 6 Conclusion

We presented CoreQ, a learning-free PTQ framework that improves layer-wise quantization under limited calibration data. CoreQ selects a corrected calibration target using a closed-form, layer-dependent correction coefficient, avoiding gradient updates, validation search, and tuning. This makes CoreQ a practical drop-in alternative to standard Hessian-based PTQ solvers: CoreQ serves as an efficient default, while $K$-CoreQ provides a search-enhanced option when additional calibration compute is available. Experiments show that adaptive mismatch correction consistently improves low-bit quantization, suggesting CoreQ as a strong default strategy for efficient LLM deployment.

**Limitations and Future Work.** While CoreQ improves over prior layer-wise PTQ methods, it remains sequential and layer-wise: the corrected calibration target accounts for errors from previously quantized layers, but CoreQ does not jointly optimize multiple layers. CoreQ also restricts calibration target selection to linear interpolation between the standard and mismatch-aware calibration targets. Developing efficient nonlinear calibration target corrections for more complex mismatches is an interesting direction for future work.

## Acknowledgment

This paper was supported in part by a DOD Award W911NF-24-2-0185 and NSF Award 2148224 and is supported in part by funds from OUSD R&E, NIST, and industry partners as specified in the Resilient & Intelligent NextG Systems (RINGS) program. We thank Kyoungpyo Lee for many insightful discussions that helped shape this work throughout its development.

## References


[1] Yingqiang Ge, Wenyue Hua, Kai Mei, Juntao Tan, Shuyuan Xu, Zelong Li, Yongfeng Zhang, et al. Openagi: When llm meets domain experts. *Advances in Neural Information Processing Systems*, 36:5539–5568, 2023.

[2] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32, 2024.

[3] Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. Using an llm to help with code understanding. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pages 1–13, 2024.

[4] Amir Gholami, Zhewei Yao, Sehoon Kim, Coleman Hooper, Michael W Mahoney, and Kurt Keutzer. Ai and memory wall. *IEEE Micro*, 44(3):33–39, 2024.

[5] 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.

[6] Jiajun Xu, Zhiyuan Li, Wei Chen, Qun Wang, Xin Gao, Qi Cai, and Ziyuan Ling. On-device language models: A comprehensive review. *arXiv preprint arXiv:2409.00088*, 2024.

[7] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International conference on machine learning*, pages 7197–7206. PMLR, 2020.

[8] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Accurate post training quantization with small calibration sets. In *International conference on machine learning*, pages 4466–4475. PMLR, 2021.

[9] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. Brecq: Pushing the limit of post-training quantization by block reconstruction. *arXiv preprint arXiv:2102.05426*, 2021.

[10] 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.

[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] Yuhang Li, Ruokai Yin, Donghyun Lee, Shiting Xiao, and Priyadarshini Panda. Gptaq: Efficient finetuning-free quantization for asymmetric calibration. In *Forty-second International Conference on Machine Learning*, 2025.

[13] Yamato Arai and Yuma Ichikawa. Quantization error propagation: Revisiting layer-wise post-training quantization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[14] Baohao Liao, Christian Herold, Shahram Khadivi, and Christof Monz. Apiq: Finetuning of 2-bit quantized large language model. *arXiv preprint arXiv:2402.05147*, 2024.

[15] Jinuk Kim, Marwa El Halabi, Wonpyo Park, Clemens JS Schaefer, Deokjae Lee, Yeonhong Park, Jae W Lee, and Hyun Oh Song. Guidedquant: Large language model quantization via exploiting end loss guidance. In *Forty-second International Conference on Machine Learning*, 2025.

[16] Albert Tseng, Zhaofeng Sun, and Christopher De Sa. Model-preserving adaptive rounding. *arXiv preprint arXiv:2505.22988*, 2025.

[17] Junhan Kim, Ho-young Kim, Eulrang Cho, Chungman Lee, Joonyoung Kim, and Yongkweon Jeon. Boa: Attention-aware post-training quantization without backpropagation. *arXiv preprint arXiv:2406.13474*, 2024.

[18] Xin Ding, Xiaoyu Liu, Zhijun Tu, Yun Zhang, Wei Li, Jie Hu, Hanting Chen, Yehui Tang, Zhiwei Xiong, Baoqun Yin, et al. Cbq: Cross-block quantization for large language models. *arXiv preprint arXiv:2312.07950*, 2023.

[19] 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.

[20] Jiale Chen, Yalda Shabanzadeh, Elvir Crnčević, Torsten Hoefler, and Dan Alistarh. The geometry of llm quantization: Gptq as babai's nearest plane algorithm. *arXiv preprint arXiv:2507.18553*, 2025.

[21] Xingyu Zheng, Haotong Qin, Yuye Li, Haoran Chu, Jiakai Wang, Jinyang Guo, Michele Magno, and Xianglong Liu. First-order error matters: Accurate compensation for quantized large language models. *arXiv preprint arXiv:2507.11017*, 2025.

[22] Kayhan Behdin, Ayan Acharya, Aman Gupta, Qingquan Song, Siyu Zhu, Sathiya Keerthi, and Rahul Mazumder. Quantease: Optimization-based quantization for language models. *arXiv preprint arXiv:2309.01885*, 2023.

[23] Pranav Ajit Nair and Arun Sai Suggala. Cdquant: Greedy coordinate descent for accurate llm quantization. *arXiv preprint arXiv:2406.17542*, 2024.

[24] Zhan Guo and Peter Nilsson. Algorithm and implementation of the k-best sphere decoding for mimo detection. *IEEE Journal on selected areas in communications*, 24(3):491–503, 2006.

[25] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

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

[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] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

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

[35] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[36] 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.

[37] Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[38] 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.

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

[40] Ali Edalati, Alireza Ghaffari, Mahsa Ghazvini Nejad, Lu Hou, Boxing Chen, Masoud Asgharian, and Vahid Partovi Nia. Oac: Output-adaptive calibration for accurate post-training quantization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16453–16461, 2025.

[41] Shihao Zhang, Haoyu Zhang, Ian Colbert, and Rayan Saab. Qronos: Correcting the past by shaping the future... in post-training quantization. In *International Conference on Learning Representations*, 2026.

[42] Haoyu Zhang, Shihao Zhang, Ian Colbert, and Rayan Saab. Provable post-training quantization: Theoretical analysis of optq and qronos. *arXiv preprint arXiv:2508.04853*, 2025.

[43] Jung Hyun Lee, Jeonghoon Kim, June Yong Yang, Se Jung Kwon, Eunho Yang, Kang Min Yoo, and Dongsoo Lee. Lrq: Optimizing post-training quantization for large language models by learning low-rank weight-scaling matrices. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7708–7743, 2025.

[44] Johann Birnick. The lattice geometry of neural network quantization: A short equivalence proof of gptq and babai's algorithm. In *International Conference on Learning Representations*, 2026.

[45] Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization. *arXiv preprint arXiv:2401.06118*, 2024.

[46] Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip $\#$: Even better llm quantization with hadamard incoherence and lattice codebooks. In *International Conference on Machine Learning*, pages 48630–48656. PMLR, 2024.

[47] Mart Van Baalen, Andrey Kuzmin, Ivan Koryakovskiy, Markus Nagel, Peter Couperus, Cedric Bastoul, Eric Mahurin, Tijmen Blankevoort, and Paul Whatmough. Gptvq: The blessing of dimensionality for llm quantization. *arXiv preprint arXiv:2402.15319*, 2024.

[48] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[49] Xing Hu, Dawei Yang, Yuan Cheng, Zhixuan Chen, and Zukang Xu. SAES-SVD: Self-adaptive suppression of accumulated and local errors for SVD-based LLM compression. In *The Fourteenth International Conference on Learning Representations*, 2026.

# A Theoretical Results

Here we provide theoretical results and explanations introduced in Section 3. We first provide a population view showing why an intermediate correction coefficient is natural.

**Notation.** Bold uppercase symbols denote deterministic matrices, including weights and finite calibration quantities. In particular, $\mathbf{X}_{\mathrm{q}}, \mathbf{X}_{\mathrm{f}} \in \mathbb{R}^{n\times N}$ denote empirical activation matrices formed from $N$ calibration activation vectors. Non-bold uppercase symbols $X_{\mathrm{q}}, X_{\mathrm{f}} \in \mathbb{R}^n$ denote the corresponding population random activation vectors. Thus, expectations such as $\mathbb{E}[X_{\mathrm{q}}X_{\mathrm{q}}^\top]$ are taken over population activations, whereas Frobenius norms involving $\mathbf{X}_{\mathrm{q}}$ or $\mathbf{X}_{\mathrm{f}}$ are empirical calibration quantities.

### A.1 Oracle shrinkage for mismatch correction

Let $X_{\mathrm{f}}, X_{\mathrm{q}} \in \mathbb{R}^n$ denote the full-precision-prefix and quantized-prefix activation random vectors at deployment, and define

$$\bar{\mathbf{H}} := \mathbb{E}[X_{\mathrm{q}}X_{\mathrm{q}}^\top], \qquad \bar{\mathbf{C}} := \mathbb{E}[X_{\mathrm{f}}X_{\mathrm{q}}^\top].$$

Assume $\bar{\mathbf{H}} \succ 0$. Writing $X_{\mathrm{f}} = X_{\mathrm{q}} + \Delta X$, define

$$\bar{\mathbf{\Delta}} := \mathbb{E}[\Delta X\, X_{\mathrm{q}}^\top], \qquad \bar{\mathbf{S}} := \mathbf{W}\bar{\mathbf{\Delta}}\bar{\mathbf{H}}^{-1}.$$

Let $\widetilde{\mathbf{S}}$ be a finite-sample calibration estimate of $\bar{\mathbf{S}}$, and consider the scalar family of continuous centers

$$\mathbf{W}^{(\alpha)} := \mathbf{W} + \alpha\widetilde{\mathbf{S}}, \qquad \alpha \in [0,1].$$

**Theorem A.1** (Oracle shrinkage for mismatch correction)**.** *The population layer-wise output error*

$$\mathcal{L}_{\mathrm{pop}}(\widehat{\mathbf{W}}) := \mathbb{E}\|\mathbf{W}X_{\mathrm{f}} - \widehat{\mathbf{W}}X_{\mathrm{q}}\|_2^2$$

*is minimized over continuous weights by*

$$\mathbf{W}^\star = \mathbf{W} + \bar{\mathbf{S}}.$$

*Assume that the finite-sample correction estimate is unbiased,* $\mathbb{E}_{\widetilde{\mathbf{S}}}[\widetilde{\mathbf{S}}] = \bar{\mathbf{S}}$*. Ignoring discrete rounding, the interpolated center* $\mathbf{W}^{(\alpha)} = \mathbf{W} + \alpha\widetilde{\mathbf{S}}$ *has the expected excess population error*

$$R(\alpha) := \mathbb{E}_{\widetilde{\mathbf{S}}}\Big[\mathcal{L}_{\mathrm{pop}}(\mathbf{W}^{(\alpha)}) - \mathcal{L}_{\mathrm{pop}}(\mathbf{W}^\star)\Big] = (1-\alpha)^2\mu^2 + \alpha^2\sigma^2,$$

*where*

$$\mu^2 := \|\bar{\mathbf{S}}\bar{\mathbf{H}}^{1/2}\|_F^2, \qquad \sigma^2 := \mathbb{E}_{\widetilde{\mathbf{S}}}\|(\widetilde{\mathbf{S}} - \bar{\mathbf{S}})\bar{\mathbf{H}}^{1/2}\|_F^2.$$

*If* $\mu^2 + \sigma^2 > 0$*, then*

$$\alpha^\star := \arg\min_{\alpha\in[0,1]} R(\alpha) = \frac{\mu^2}{\mu^2 + \sigma^2}.$$

*In particular,* $0 < \alpha^\star < 1$ *whenever* $\mu^2 > 0$ *and* $\sigma^2 > 0$*.*

*Proof.* Expanding the population loss gives

$$\mathcal{L}_{\mathrm{pop}}(\widehat{\mathbf{W}}) = \mathrm{tr}\big(\mathbf{W}\mathbb{E}[X_{\mathrm{f}}X_{\mathrm{f}}^\top]\mathbf{W}^\top\big) - 2\,\mathrm{tr}\Big(\mathbf{W}\bar{\mathbf{C}}\widehat{\mathbf{W}}^\top\Big) + \mathrm{tr}\Big(\widehat{\mathbf{W}}\bar{\mathbf{H}}\widehat{\mathbf{W}}^\top\Big).$$

Completing the square gives

$$\mathcal{L}_{\mathrm{pop}}(\widehat{\mathbf{W}}) - \mathcal{L}_{\mathrm{pop}}(\mathbf{W}^\star) = \|(\widehat{\mathbf{W}} - \mathbf{W}^\star)\bar{\mathbf{H}}^{1/2}\|_F^2.$$

Since $X_{\mathrm{f}} = X_{\mathrm{q}} + \Delta X$,

$$\bar{\mathbf{C}} = \bar{\mathbf{H}} + \bar{\mathbf{\Delta}},$$

so

$$\mathbf{W}\bar{\mathbf{C}}\bar{\mathbf{H}}^{-1} = \mathbf{W}(\bar{\mathbf{H}} + \bar{\mathbf{\Delta}})\bar{\mathbf{H}}^{-1} = \mathbf{W} + \mathbf{W}\bar{\mathbf{\Delta}}\bar{\mathbf{H}}^{-1} = \mathbf{W} + \bar{\mathbf{S}}.$$

Thus $\mathbf{W}^\star = \mathbf{W} + \bar{\mathbf{S}}$.

For the interpolated center $\mathbf{W}^{(\alpha)} = \mathbf{W} + \alpha\widetilde{\mathbf{S}}$,

$$\mathbf{W}^{(\alpha)} - \mathbf{W}^\star = \mathbf{W} + \alpha\widetilde{\mathbf{S}} - (\mathbf{W} + \bar{\mathbf{S}}) = -(1-\alpha)\bar{\mathbf{S}} + \alpha(\widetilde{\mathbf{S}} - \bar{\mathbf{S}}).$$

Therefore,

$$\begin{aligned} R(\alpha) &= \mathbb{E}_{\widetilde{\mathbf{S}}}\left[\|(\mathbf{W}^{(\alpha)} - \mathbf{W}^\star)\bar{\mathbf{H}}^{1/2}\|_F^2\right] \\ &= (1-\alpha)^2\|\bar{\mathbf{S}}\bar{\mathbf{H}}^{1/2}\|_F^2 + \alpha^2\mathbb{E}_{\widetilde{\mathbf{S}}}\|(\widetilde{\mathbf{S}} - \bar{\mathbf{S}})\bar{\mathbf{H}}^{1/2}\|_F^2, \end{aligned}$$

where the cross term vanishes because $\mathbb{E}_{\widetilde{\mathbf{S}}}[\widetilde{\mathbf{S}} - \bar{\mathbf{S}}] = 0$. Minimizing the resulting scalar quadratic gives

$$\alpha^\star = \frac{\mu^2}{\mu^2 + \sigma^2}.$$

□

Theorem A.1 explains why interpolation is needed: $\alpha = 0$ ignores the population correction $\bar{\mathbf{S}}$, while $\alpha = 1$ fully trusts the finite-sample estimate $\widetilde{\mathbf{S}}$. The optimal scalar coefficient balances population correction energy $\mu^2$ against finite-calibration estimation error $\sigma^2$, and is therefore intermediate whenever both are present.

### A.2 Connection to the empirical criterion

Let $\mathcal{R} := \{\mathbf{A}\mathbf{X}_\mathrm{q} : \mathbf{A} \in \mathbb{R}^{m\times n}\}$ and $\mathbf{D} := \mathbf{W}(\mathbf{X}_\mathrm{f} - \mathbf{X}_\mathrm{q})$. The following decomposition uses the orthogonal projection of $\mathbf{D}$ onto $\mathcal{R}$, so that $\mathbf{s} = \widetilde{\mathbf{S}}\mathbf{X}_\mathrm{q} \in \mathcal{R}$ and $\boldsymbol{\eta} \perp \mathcal{R}$.

**Lemma A.2** (Reliability-regularized empirical target risk)**.** *Let*

$$\mathbf{W}^{(\alpha)} := \mathbf{W} + \alpha\widetilde{\mathbf{S}}, \qquad \mathbf{Y}_\mathcal{R} := (\mathbf{W} + \widetilde{\mathbf{S}})\mathbf{X}_\mathrm{q}, \qquad \mathbf{D} = \mathbf{s} + \boldsymbol{\eta}, \qquad \mathbf{s} := \widetilde{\mathbf{S}}\mathbf{X}_\mathrm{q}, \quad \boldsymbol{\eta} \perp \mathcal{R}.$$

*Let*

$$\mathbf{X}^{(\alpha)} := (1-\alpha)\mathbf{X}_\mathrm{q} + \alpha\mathbf{X}_\mathrm{f}.$$

*The empirical plug-in analogue of the excess population error in Theorem A.1 is*

$$\widehat{R}_\mathrm{plug}(\alpha) := \|(\mathbf{W}^{(\alpha)} - (\mathbf{W} + \widetilde{\mathbf{S}}))\mathbf{X}_\mathrm{q}\|_F^2 = (1-\alpha)^2\|\mathbf{s}\|_F^2.$$

*Moreover, the displacement criterion used in CoreQ satisfies*

$$\|\mathbf{Y}_\mathcal{R} - \mathbf{W}\mathbf{X}^{(\alpha)}\|_F^2 = \widehat{R}_\mathrm{plug}(\alpha) + \|\mathbf{W}\mathbf{X}^{(\alpha)} - \mathbf{W}^{(\alpha)}\mathbf{X}_\mathrm{q}\|_F^2 = (1-\alpha)^2\|\mathbf{s}\|_F^2 + \alpha^2\|\boldsymbol{\eta}\|_F^2.$$

*Proof.* Since $\mathbf{W}^{(\alpha)}\mathbf{X}_\mathrm{q} = \mathbf{W}\mathbf{X}_\mathrm{q} + \alpha\mathbf{s}$ and $\mathbf{Y}_\mathcal{R} = \mathbf{W}\mathbf{X}_\mathrm{q} + \mathbf{s}$,

$$\widehat{R}_\mathrm{plug}(\alpha) = \|\mathbf{Y}_\mathcal{R} - \mathbf{W}^{(\alpha)}\mathbf{X}_\mathrm{q}\|_F^2 = (1-\alpha)^2\|\mathbf{s}\|_F^2.$$

Also,

$$\mathbf{W}\mathbf{X}^{(\alpha)} = \mathbf{W}\mathbf{X}_\mathrm{q} + \alpha\mathbf{D} = \mathbf{W}\mathbf{X}_\mathrm{q} + \alpha\mathbf{s} + \alpha\boldsymbol{\eta} = \mathbf{W}^{(\alpha)}\mathbf{X}_\mathrm{q} + \alpha\boldsymbol{\eta}.$$

Therefore

$$\mathbf{Y}_\mathcal{R} - \mathbf{W}\mathbf{X}^{(\alpha)} = (\mathbf{Y}_\mathcal{R} - \mathbf{W}^{(\alpha)}\mathbf{X}_\mathrm{q}) + (\mathbf{W}^{(\alpha)}\mathbf{X}_\mathrm{q} - \mathbf{W}\mathbf{X}^{(\alpha)}) = (1-\alpha)\mathbf{s} - \alpha\boldsymbol{\eta}.$$

The two terms are orthogonal because $\mathbf{s} \in \mathcal{R}$ and $\boldsymbol{\eta} \perp \mathcal{R}$, giving the stated decomposition. □

Theorem A.1 shows that, when the finite-calibration correction estimate is imperfect, the desired correction need not be full and the population-optimal coefficient balances the benefit of applying the correction against the cost of trusting an inaccurate estimate. This motivates shrinkage, but it does not by itself provide a computable coefficient, since the population correction energy and estimation error are unavailable in layer-wise PTQ.

The calibration-set analogue in Lemma A.2 measures only the distance to the finite-sample reachable target and therefore always prefers $\alpha = 1$. This is overly optimistic because it provides no penalty for the portion of the observed mismatch that cannot be produced by changing the current layer's weights.

CoreQ therefore uses the orthogonal residual energy $\|\boldsymbol{\eta}\|_F^2$ as an observable geometric penalty for such unreachable mismatch. This yields the reliability-regularized distance metric

$$\|\mathbf{d}(\alpha)\|_F^2 = \|\mathbf{Y}_{\mathcal{R}} - \mathbf{W}\mathbf{X}^{(\alpha)}\|_F^2 = (1-\alpha)^2\|\mathbf{s}\|_F^2 + \alpha^2\|\boldsymbol{\eta}\|_F^2,$$

whose minimizer is

$$\alpha_{\mathrm{corr}} = \frac{\|\mathbf{s}\|_F^2}{\|\mathbf{s}\|_F^2 + \|\boldsymbol{\eta}\|_F^2}.$$

Thus, $\alpha_{\mathrm{corr}}$ measures the empirical fraction of observed mismatch energy that lies in the reachable response space of the current layer. It shrinks the correction when a large portion of the mismatch is structurally unexplained by the current layer, which helps avoid over-trusting calibration-specific mismatch patterns.

### A.3 Extension to biased correction estimators

The unbiasedness assumption in Theorem A.1 need not hold for an estimator involving a random matrix inverse. We first derive the expected excess population error and optimal correction coefficient when the calibration estimator may be biased. We then bound the deviation from the coefficient given by the unbiased formula and the additional risk incurred by retaining that formula. Finally, we use an exact identity for the correction estimator to relate these bounds to calibration size, establishing an $O(N^{-1})$ coefficient gap and an $O(N^{-2})$ risk penalty under the stated sampling, moment, and stability assumptions. Throughout the analysis, the layer weights and prefixes remain fixed across repeated calibration draws.

#### A.3.1 Population risk with estimator bias

Let $\mathbf{X}_{\mathrm{f}}, \mathbf{X}_{\mathrm{q}} \in \mathbb{R}^{n\times N}$ denote the full-precision and quantized-prefix activation matrices for a randomly drawn calibration set of size $N$. Define the corresponding output-mismatch matrix

$$\mathbf{D} := \mathbf{W}(\mathbf{X}_{\mathrm{f}} - \mathbf{X}_{\mathrm{q}}) \in \mathbb{R}^{m\times N}.$$

The normalized empirical moments and finite-calibration correction are

$$\widehat{\mathbf{H}}_N := \frac{1}{N}\mathbf{X}_{\mathrm{q}}\mathbf{X}_{\mathrm{q}}^\top, \qquad \widehat{\mathbf{M}}_N := \frac{1}{N}\mathbf{D}\mathbf{X}_{\mathrm{q}}^\top, \qquad \widetilde{\mathbf{S}}_N := \widehat{\mathbf{M}}_N\widehat{\mathbf{H}}_N^{-1}.$$

The population moments and population correction are

$$\bar{\mathbf{H}} := \mathbb{E}[\widehat{\mathbf{H}}_N] \succ 0, \qquad \bar{\mathbf{M}} := \mathbb{E}[\widehat{\mathbf{M}}_N], \qquad \bar{\mathbf{S}} := \bar{\mathbf{M}}\bar{\mathbf{H}}^{-1}.$$

All expectations are over repeated random calibration-set draws. With the population moments defined above, $\bar{\mathbf{M}} = \mathbf{W}(\bar{\mathbf{C}} - \bar{\mathbf{H}})$, and equivalently $\bar{\mathbf{S}} = \mathbf{W}(\bar{\mathbf{C}}\bar{\mathbf{H}}^{-1} - \mathbf{I})$.

For matrices $\mathbf{A}$ and $\mathbf{B}$, define the $\bar{\mathbf{H}}$-weighted inner product

$$\langle \mathbf{A}, \mathbf{B}\rangle_{\bar{\mathbf{H}}} := \mathrm{tr}(\mathbf{A}\bar{\mathbf{H}}\mathbf{B}^\top), \qquad \|\mathbf{A}\|_{\bar{\mathbf{H}}}^2 := \langle \mathbf{A}, \mathbf{A}\rangle_{\bar{\mathbf{H}}} = \|\mathbf{A}\bar{\mathbf{H}}^{1/2}\|_F^2.$$

Define the finite-sample estimation error and its mean bias by

$$\mathbf{E}_N := \widetilde{\mathbf{S}}_N - \bar{\mathbf{S}}, \qquad \mathbf{B}_N := \mathbb{E}[\mathbf{E}_N] = \mathbb{E}[\widetilde{\mathbf{S}}_N] - \bar{\mathbf{S}}.$$

Also define

$$\mu^2 := \|\bar{\mathbf{S}}\bar{\mathbf{H}}^{1/2}\|_F^2, \qquad \sigma_N^2 := \mathbb{E}\|\mathbf{E}_N\bar{\mathbf{H}}^{1/2}\|_F^2,$$

and the correction-aligned bias

$$\rho_N := \langle \bar{\mathbf{S}}, \mathbf{B}_N\rangle_{\bar{\mathbf{H}}} = \mathrm{tr}(\bar{\mathbf{S}}\bar{\mathbf{H}}\mathbf{B}_N^\top).$$

Here $\sigma_N^2$ is the mean-square estimation error around $\bar{\mathbf{S}}$, which includes both variance and squared bias.

**Theorem A.3** (Oracle coefficient with estimator bias)**.** *By the completed-square identity in the proof of Theorem A.1, the expected excess population error for* $\mathbf{W} + \alpha\widetilde{\mathbf{S}}_N$ *is*

$$R_N(\alpha) := \mathbb{E}\left\|(\alpha\widetilde{\mathbf{S}}_N - \bar{\mathbf{S}})\bar{\mathbf{H}}^{1/2}\right\|_F^2.$$

*Then*

$$R_N(\alpha) = (1-\alpha)^2\mu^2 + \alpha^2\sigma_N^2 - 2\alpha(1-\alpha)\rho_N.$$

*If* $\mu^2 + \sigma_N^2 + 2\rho_N > 0$*, the unconstrained minimizer is*

$$\alpha^\star_{N,\mathrm{bias}} = \frac{\mu^2 + \rho_N}{\mu^2 + \sigma_N^2 + 2\rho_N}.$$

*The constrained minimizer over* $[0,1]$ *is obtained by projecting this value onto* $[0,1]$.

*Proof.* Since

$$\alpha\widetilde{\mathbf{S}}_N - \bar{\mathbf{S}} = -(1-\alpha)\bar{\mathbf{S}} + \alpha\mathbf{E}_N,$$

expanding the squared $\bar{\mathbf{H}}$-norm gives

$$R_N(\alpha) = (1-\alpha)^2\|\bar{\mathbf{S}}\|^2_{\bar{\mathbf{H}}} + \alpha^2\mathbb{E}\|\mathbf{E}_N\|^2_{\bar{\mathbf{H}}} - 2\alpha(1-\alpha)\mathbb{E}\langle\bar{\mathbf{S}}, \mathbf{E}_N\rangle_{\bar{\mathbf{H}}}.$$

The three terms are $\mu^2$, $\sigma_N^2$, and $\rho_N$, respectively. Differentiating the resulting quadratic yields the stated minimizer. □

Theorem A.1 assumes $\mathbb{E}[\widetilde{\mathbf{S}}_N] = \bar{\mathbf{S}}$, which implies $\mathbf{B}_N = 0$ and hence $\rho_N = 0$. The bias-aware theorem shows that the original coefficient remains exact under the weaker condition $\rho_N = 0$, even when $\mathbf{B}_N \neq 0$. Only the aligned mean bias contributes to the cross term; all bias components are also accounted for through $\sigma_N^2$.

Let

$$M_N := \mu^2 + \sigma_N^2, \qquad \alpha^\star_{N,0} := \frac{\mu^2}{M_N},$$

where $M_N > 0$ and $\alpha^\star_{N,0}$ is the coefficient from the unbiased formula. When $M_N + 2\rho_N > 0$, a direct calculation gives

$$\alpha^\star_{N,\mathrm{bias}} - \alpha^\star_{N,0} = \frac{\rho_N(\sigma_N^2 - \mu^2)}{M_N(M_N + 2\rho_N)}.$$

**Lemma A.4** (Robustness to aligned bias)**.** *Suppose*

$$|\rho_N| \le \frac{\epsilon_N}{2}M_N, \qquad 0 \le \epsilon_N < 1.$$

*Then*

$$\left|\alpha^\star_{N,\mathrm{bias}} - \alpha^\star_{N,0}\right| \le \frac{\epsilon_N}{2(1-\epsilon_N)}.$$

*Moreover,*

$$R_N(\alpha^\star_{N,0}) - R_N(\alpha^\star_{N,\mathrm{bias}}) \le \frac{\epsilon_N^2}{4(1-\epsilon_N)}M_N.$$

*The same bounds remain valid after projection of the biased optimum onto* $[0,1]$.

*Proof.* Because $|\sigma_N^2 - \mu^2| \le M_N$ and $M_N + 2\rho_N \ge (1-\epsilon_N)M_N$,

$$\left|\alpha^\star_{N,\mathrm{bias}} - \alpha^\star_{N,0}\right| \le \frac{|\rho_N|}{(1-\epsilon_N)M_N} \le \frac{\epsilon_N}{2(1-\epsilon_N)}.$$

The risk is a quadratic with curvature $M_N + 2\rho_N$, so

$$R_N(\alpha) - R_N(\alpha^\star_{N,\mathrm{bias}}) = (M_N + 2\rho_N)(\alpha - \alpha^\star_{N,\mathrm{bias}})^2.$$

Using the exact coefficient difference and the lower bound on the denominator produces the stated risk bound. Projection cannot increase the distance from $\alpha^\star_{N,0} \in [0,1]$ or decrease the constrained minimum below the unconstrained minimum. □

A small aligned bias perturbs the optimal coefficient at first order in $\epsilon_N$, whereas the excess-risk cost of retaining the original coefficient is second order in $\epsilon_N$. Thus, exact matrix unbiasedness is not required for the original theorem to remain practically accurate.

### A.3.2 Dependence on calibration size

We next quantify how $\rho_N$ depends on the calibration size. The argument separates two facts: (i) empirical moments fluctuate at order $N^{-1/2}$, and (ii) the ratio-estimator bias arises from products of these centered moment errors, hence at second order.

Define the whitened empirical-moment errors

$$\mathbf{G}_N := \bar{\mathbf{H}}^{-1/2}(\widehat{\mathbf{H}}_N - \bar{\mathbf{H}})\bar{\mathbf{H}}^{-1/2}, \qquad \mathbf{F}_N := (\widehat{\mathbf{M}}_N - \bar{\mathbf{M}})\bar{\mathbf{H}}^{-1/2},$$

and let

$$\mathbf{A} := \bar{\mathbf{S}}\bar{\mathbf{H}}^{1/2}.$$

Then

$$\widehat{\mathbf{H}}_N = \bar{\mathbf{H}}^{1/2}(\mathbf{I} + \mathbf{G}_N)\bar{\mathbf{H}}^{1/2}.$$

**Lemma A.5** (Exact ratio-estimator identity)**.** *Whenever $\widehat{\mathbf{H}}_N$ is invertible,*

$$(\widetilde{\mathbf{S}}_N - \bar{\mathbf{S}})\bar{\mathbf{H}}^{1/2} = (\mathbf{F}_N - \mathbf{A}\mathbf{G}_N)(\mathbf{I} + \mathbf{G}_N)^{-1}.$$

*Proof.* Since

$$\widehat{\mathbf{H}}_N^{-1} = \bar{\mathbf{H}}^{-1/2}(\mathbf{I} + \mathbf{G}_N)^{-1}\bar{\mathbf{H}}^{-1/2}$$

and

$$\widehat{\mathbf{M}}_N\bar{\mathbf{H}}^{-1/2} = \mathbf{A} + \mathbf{F}_N,$$

we have

$$\widetilde{\mathbf{S}}_N\bar{\mathbf{H}}^{1/2} = (\mathbf{A} + \mathbf{F}_N)(\mathbf{I} + \mathbf{G}_N)^{-1}.$$

Subtracting $\mathbf{A} = \bar{\mathbf{S}}\bar{\mathbf{H}}^{1/2}$ gives

$$(\mathbf{A} + \mathbf{F}_N)(\mathbf{I} + \mathbf{G}_N)^{-1} - \mathbf{A} = (\mathbf{F}_N - \mathbf{A}\mathbf{G}_N)(\mathbf{I} + \mathbf{G}_N)^{-1}.$$

□

To establish the $1/N$ rate, we impose the following assumptions.

**Assumption A.6** (Independent sampling units)**.** *The $N$ columns of $(\mathbf{X}_{\mathrm{f}}, \mathbf{X}_{\mathrm{q}})$ are independent and identically distributed paired sampling units. Equivalently, if $\mathbf{x}_{\mathrm{q},i}$ and $\mathbf{d}_i$ denote the $i$th columns of $\mathbf{X}_{\mathrm{q}}$ and $\mathbf{D}$, then $\{(\mathbf{d}_i, \mathbf{x}_{\mathrm{q},i})\}_{i=1}^N$ are i.i.d.*

**Assumption A.7** (Finite moment variances)**.** *The centered single-sample moment contributions*

$$\mathbf{G}_i := \bar{\mathbf{H}}^{-1/2}(\mathbf{x}_{\mathrm{q},i}\mathbf{x}_{\mathrm{q},i}^\top - \bar{\mathbf{H}})\bar{\mathbf{H}}^{-1/2},$$

$$\mathbf{F}_i := (\mathbf{d}_i\mathbf{x}_{\mathrm{q},i}^\top - \bar{\mathbf{M}})\bar{\mathbf{H}}^{-1/2}$$

*have finite second moments. Define*

$$v_H := \mathbb{E}\|\mathbf{G}_i\|_F^2 < \infty, \qquad v_M := \mathbb{E}\|\mathbf{F}_i\|_F^2 < \infty.$$

**Assumption A.8** (Uniform covariance stability)**.** *For some $\delta \in [0, 1)$,*

$$\|\mathbf{G}_N\|_2 \le \delta$$

*almost surely over calibration draws. Equivalently, the empirical covariance satisfies*

$$(1 - \delta)\bar{\mathbf{H}} \preceq \widehat{\mathbf{H}}_N \preceq (1 + \delta)\bar{\mathbf{H}}.$$

Assumptions A.6 and A.7 justify the $1/N$ moment scaling rather than merely postulating it. Indeed,

$$\mathbf{F}_N = \frac{1}{N}\sum_{i=1}^N \mathbf{F}_i, \qquad \mathbf{G}_N = \frac{1}{N}\sum_{i=1}^N \mathbf{G}_i,$$

with $\mathbb{E}[\mathbf{F}_i] = \mathbb{E}[\mathbf{G}_i] = 0$. Independence eliminates all cross terms, so

$$\mathbb{E}\|\mathbf{F}_N\|_F^2 = \frac{v_M}{N}, \qquad \mathbb{E}\|\mathbf{G}_N\|_2^2 \le \mathbb{E}\|\mathbf{G}_N\|_F^2 = \frac{v_H}{N}.$$

The covariance-stability assumption is separate: it controls the random inverse. Under bounded or sub-Gaussian sampling units, standard covariance concentration makes this event high probability once the calibration size is large relative to the effective activation dimension.

**Theorem A.9** (Finite calibration bias bound)**.** *Under assumptions A.6–A.8,*

$$\|\mathbf{B}_N\bar{\mathbf{H}}^{1/2}\|_F \le \frac{\sqrt{v_M v_H}+\mu v_H}{(1-\delta)N}, \qquad \mu := \|\bar{\mathbf{S}}\bar{\mathbf{H}}^{1/2}\|_F.$$

*Consequently,*

$$|\rho_N| \le \frac{\mu\big(\sqrt{v_M v_H}+\mu v_H\big)}{(1-\delta)N}.$$

*Proof.* Let $\mathbf{Q}_N := \mathbf{F}_N - \mathbf{A}\mathbf{G}_N$. Since $\mathbb{E}[\mathbf{F}_N] = \mathbb{E}[\mathbf{G}_N] = 0$, $\mathbb{E}[\mathbf{Q}_N] = 0$. Using

$$\mathbf{Q}_N(\mathbf{I}+\mathbf{G}_N)^{-1} = \mathbf{Q}_N - \mathbf{Q}_N\mathbf{G}_N(\mathbf{I}+\mathbf{G}_N)^{-1},$$

the exact identity implies

$$\mathbf{B}_N\bar{\mathbf{H}}^{1/2} = -\mathbb{E}\left[\mathbf{Q}_N\mathbf{G}_N(\mathbf{I}+\mathbf{G}_N)^{-1}\right].$$

Assumption A.8 gives $\|(\mathbf{I}+\mathbf{G}_N)^{-1}\|_2 \le (1-\delta)^{-1}$. Therefore,

$$\begin{aligned}\|\mathbf{B}_N\bar{\mathbf{H}}^{1/2}\|_F &\le \frac{1}{1-\delta}\mathbb{E}\left[(\|\mathbf{F}_N\|_F + \mu\|\mathbf{G}_N\|_2)\|\mathbf{G}_N\|_2\right] \\ &\le \frac{1}{1-\delta}\left(\sqrt{\mathbb{E}\|\mathbf{F}_N\|_F^2\,\mathbb{E}\|\mathbf{G}_N\|_2^2} + \mu\mathbb{E}\|\mathbf{G}_N\|_2^2\right) \\ &\le \frac{\sqrt{v_M v_H}+\mu v_H}{(1-\delta)N}.\end{aligned}$$

Finally, weighted Cauchy–Schwarz yields

$$|\rho_N| = |\langle\bar{\mathbf{S}}, \mathbf{B}_N\rangle_{\bar{\mathbf{H}}}| \le \|\bar{\mathbf{S}}\bar{\mathbf{H}}^{1/2}\|_F\|\mathbf{B}_N\bar{\mathbf{H}}^{1/2}\|_F,$$

which gives the second bound. □

The constants $v_H$ and $v_M$ contain the dimensional and distributional dependence. If $\mathbf{u}_i := \bar{\mathbf{H}}^{-1/2}\mathbf{x}_{\mathrm{q},i} \in \mathbb{R}^n$ is the whitened activation, then

$$v_H = \mathbb{E}\|\mathbf{u}_i\mathbf{u}_i^\top - \mathbf{I}\|_F^2 = \mathbb{E}\|\mathbf{u}_i\|_2^4 - n.$$

Thus, $v_H$ reflects the activation dimension and its fourth moments. Similarly,

$$v_M = \mathbb{E}\left\|(\mathbf{d}_i\mathbf{x}_{\mathrm{q},i}^\top - \bar{\mathbf{M}})\bar{\mathbf{H}}^{-1/2}\right\|_F^2$$

depends on the output dimension, activation dimension, and mixed fourth moments of the mismatch and activations. The constants therefore retain the dimension and distribution dependence of the moment fluctuations.

Define

$$\epsilon_N := \frac{2|\rho_N|}{M_N}, \qquad M_N := \mu^2 + \sigma_N^2.$$

If $M_N$ is bounded away from zero and the assumptions above hold with constants independent of $N$, then

$$\epsilon_N = O(N^{-1}).$$

Lemma A.4 therefore gives

$$\left|\alpha^\star_{N,\mathrm{bias}} - \alpha^\star_{N,0}\right| = O(N^{-1}),$$

while

$$R_N(\alpha^\star_{N,0}) - R_N(\alpha^\star_{N,\mathrm{bias}}) = O(N^{-2}).$$

Hence the ratio-estimator bias may perturb the optimal coefficient at order $1/N$, but the resulting excess-risk penalty from retaining the original coefficient is second order, at order $1/N^2$.

## B Rounding Algorithms

**Notation.** In this appendix, bold uppercase symbols denote matrices. In particular, $\mathbf{W} \in \mathbb{R}^{m\times n}$ denotes the full-precision weight, $\mathbf{W}_{\text{corr}} := \mathbf{W} + \alpha_{\text{corr}}\widetilde{\mathbf{S}}$ denotes the mismatch-corrected continuous weight (or rounding center), $\mathbf{Q} \in \mathbb{R}^{m\times n}$ denotes the quantized weight, $\mathbf{H} \in \mathbb{R}^{n\times n}$ denotes the Hessian proxy, and $\mathbf{L}$ its Cholesky factor. We use $\mathbf{Q}_{:,j}$ for column $j$, $\mathbf{Q}_{i,:}$ for row $i$, and keep scalar coordinates such as $q_j$, $c_j$, and $\Delta_j$ non-bold.

We provide pseudocode for CoreQ and $K$-CoreQ, together with practical implementation details that bridge the objective reformulation and search procedures used in our codebase. Specifically, our algorithm for computing $\alpha_{\text{corr}}$ is provided in Algorithm 1. Algorithm 2 presents the CoreQ algorithm with lazy-batch implementation [11]. Building on the same triangular proxy, Algorithm 3 extends CoreQ to a $K$-best beam search that maintains multiple candidates per output row and enables a bounded-complexity improvement over greedy rounding.

To make the connection explicit, we first provide a matrix-level interpretation that shows how the Cholesky-triangular form yields a columnwise branch-metric decomposition with an interference-cancelled center (Proposition B.1), which directly motivates successive rounding and its beam-search generalization. We then detail our beam-search implementation.

### B.1 Matrix-level interpretation of objective reformulation and successive rounding

**Proposition B.1** (Columnwise decomposition of the triangular proxy)**.** *Let $\mathbf{H} \in \mathbb{R}^{n\times n}$ be SPD and admit a Cholesky factorization $\mathbf{H} = \mathbf{L}\mathbf{L}^\top$, where $\mathbf{L}$ is lower triangular and $\mathbf{L}_{j,j} > 0$. Define $\mathbf{E} := \mathbf{Q} - \mathbf{W}_{\text{corr}} \in \mathbb{R}^{m\times n}$. Given the calibration objective*

$$\mathcal{L}(\mathbf{Q}) = \|(\mathbf{Q} - \mathbf{W}_{\text{corr}})\mathbf{L}\|_F^2 = \|\mathbf{E}\mathbf{L}\|_F^2,$$

*we have*

$$\mathcal{L}(\mathbf{Q}) = \sum_{j=1}^{n} \mathbf{L}_{j,j}^2 \left\| \mathbf{E}_{:,j} + \sum_{k=j+1}^{n} \mathbf{E}_{:,k} \frac{\mathbf{L}_{k,j}}{\mathbf{L}_{j,j}} \right\|_2^2. \tag{12}$$

*Equivalently, defining*

$$\widetilde{\mathbf{W}}_{:,j} := \mathbf{W}_{\text{corr},:,j} + \sum_{k=j+1}^{n} (\mathbf{W}_{\text{corr},:,k} - \mathbf{Q}_{:,k}) \frac{\mathbf{L}_{k,j}}{\mathbf{L}_{j,j}}, \tag{13}$$

*we have the diagonalized form*

$$\mathcal{L}(\mathbf{Q}) = \sum_{j=1}^{n} \mathbf{L}_{j,j}^2 \, \|\widetilde{\mathbf{W}}_{:,j} - \mathbf{Q}_{:,j}\|_2^2. \tag{14}$$

Equation (14) is precisely the branch-metric decomposition that appears in sphere decoding: $\mathbf{L}_{j,j}^2$ acts as a per-level weight, and $\widetilde{\mathbf{W}}_{:,j}$ is the interference-cancelled center for column $j$ given already-chosen later columns $\{\mathbf{Q}_{:,k}\}_{k>j}$. Thus, choosing $\mathbf{Q}$ corresponds to *traversing a tree* over columns $j = n, n-1, \ldots, 1$, where each decision fixes a column and updates the targets for earlier columns.

*Proof.* Let $\mathbf{e}_j$ denote the $j$-th standard basis vector in $\mathbb{R}^n$. The $j$-th column of $\mathbf{E}\mathbf{L}$ is

$$(\mathbf{E}\mathbf{L})_{:,j} = \mathbf{E}\mathbf{L}_{:,j} = \sum_{k=1}^{n} \mathbf{E}_{:,k}\mathbf{L}_{k,j}.$$

Since $\mathbf{L}$ is lower triangular, $\mathbf{L}_{k,j} = 0$ for $k < j$, hence

$$(\mathbf{E}\mathbf{L})_{:,j} = \sum_{k=j}^{n} \mathbf{E}_{:,k}\mathbf{L}_{k,j} = \mathbf{L}_{j,j} \left( \mathbf{E}_{:,j} + \sum_{k=j+1}^{n} \mathbf{E}_{:,k} \frac{\mathbf{L}_{k,j}}{\mathbf{L}_{j,j}} \right).$$

Taking squared $\ell_2$ norms and summing over $j$ yields

$$\|\mathbf{EL}\|_F^2 = \sum_{j=1}^{n} \|(\mathbf{EL})_{:,j}\|_2^2 = \sum_{j=1}^{n} \mathbf{L}_{j,j}^2 \left\| \mathbf{E}_{:,j} + \sum_{k=j+1}^{n} \mathbf{E}_{:,k} \frac{\mathbf{L}_{k,j}}{\mathbf{L}_{j,j}} \right\|_2^2, \tag{15}$$

which proves (12). Substituting $\mathbf{E} = \mathbf{Q} - \mathbf{W}_{\text{corr}}$ and rearranging gives

$$\widetilde{\mathbf{W}}_{:,j} - \mathbf{Q}_{:,j} = -\left( \mathbf{E}_{:,j} + \sum_{k>j} \mathbf{E}_{:,k} \frac{\mathbf{L}_{k,j}}{\mathbf{L}_{j,j}} \right),$$

and thus (14). □

**Permutation matrix P.** The reformulation in (14) also motivates a column permutation of the weight matrix prior to successive rounding. In (14), the proxy decomposes into level-wise contributions $\mathbf{L}_{j,j}^2 \|\widetilde{\mathbf{W}}_{:,j} - \mathbf{Q}_{:,j}\|_2^2$, where the weight $\mathbf{L}_{j,j}^2$ reflects the local scaling induced by $\mathbf{H}$. Since $\widetilde{\mathbf{W}}_{:,j}$ is an interference-cancelled center that depends on the already-fixed columns $\{\mathbf{Q}_{:,k}\}_{k>j}$, the overall procedure is order-dependent: decisions made first establish the tail $\{\mathbf{Q}_{:,k}\}_{k>j}$ that conditions all subsequent columns.

Accordingly, we define a permutation $\pi : [n] \to [n]$ that sorts coordinates as

$$\mathbf{H}_{\pi(1),\pi(1)} \leq \mathbf{H}_{\pi(2),\pi(2)} \leq \cdots \leq \mathbf{H}_{\pi(n),\pi(n)}.$$

We then apply this permutation to reorder the columns before running successive rounding with a right-to-left traversal $j = n, n-1, \ldots, 1$. Under this convention, the coordinates with larger diagonal curvature $\mathbf{H}_{ii}$ are placed closer to the beginning of the decoding order and are therefore quantized first. This permutation is used throughout all of our implementations.

We note that this interpretation is exact only in the diagonal case: if $\mathbf{H}$ is diagonal, then its Cholesky factor satisfies $\mathbf{L} = \operatorname{diag}\left(\sqrt{\mathbf{H}_{11}}, \ldots, \sqrt{\mathbf{H}_{nn}}\right)$, so sorting by $\mathbf{H}_{ii}$ is equivalent to sorting by the true per-level weights $\mathbf{L}_{j,j}^2$ in (14). When $\mathbf{H}$ is not diagonal, the diagonal entries $\mathbf{H}_{ii}$ no longer uniquely determine $\mathbf{L}_{j,j}$ due to off-diagonal couplings. Nevertheless, in practice we find that ordering columns by the diagonal proxy $\mathbf{H}_{ii}$ yields a consistent and effective decoding order, serving as a simple curvature-based heuristic that remains well aligned with the dominant directions even in the presence of correlations.

### B.2 Beam Search Implementation

At a given level, or column, $j = n, \ldots, 1$, for each row and beam, we form the interference-cancelled center using the triangular couplings. In the columnwise batched form, this corresponds to the $\widetilde{\mathbf{W}}_{:,j}$ update in (13). In our code, we compute the same quantity as a beam-dependent tensor $\widetilde{\mathbf{W}} \in \mathbb{R}^{m \times K}$ via batched matrix multiplications.

Given $\widetilde{\mathbf{W}}$ for column $j$, we evaluate candidate quantization levels $q_j \in \mathcal{Q}_j$ by broadcasting and compute the incremental costs $\mathbf{\Delta}_j = \mathbf{L}_{j,j}^2 \, |q_j - \widetilde{\mathbf{W}}|^2 \in \mathbb{R}^{m \times K \times A}$. We then update scores $\mathbf{S}' = \mathbf{S} + \mathbf{\Delta}_j$, broadcast over the $A$ candidate levels, flatten the beam-and-choice axis, and keep the best $K$ per row via `topk`, letting us update the tail state $\mathbf{Q}_{\text{tail}}$. This implements an exact $K$-best search under the triangular branch metrics while remaining GPU-friendly. Algorithm 3 shows the $K$-CoreQ algorithm with lazy-batch update, and the dimensions of matrices and tensors used are presented in Table 5.

For clarity, we define the three auxiliary operators used in Algorithm 3. All of them operate row-wise, independently for each output row $r \in [m]$, since the proxy objective decomposes over rows.

**TopK$(\mathbf{J}, K)$.** Let $\mathbf{J} \in \mathbb{R}^{m \times K \times A}$ be a tensor of candidate costs, where $K$ indexes the current beam and $A$ indexes the discrete quantization levels. For each row $r$, define the set of all beam–level pairs $\Omega = \{(b, a) : b \in [K],\ a \in [A]\}$, and the corresponding costs $\rho_r(b, a) := \mathbf{J}_{r,b,a}$. TopK returns the $K$ smallest costs and their argmin indices:

$$\{(b_{r,k}^\star, a_{r,k}^\star)\}_{k=1}^{K} = \text{KSmallest}\big(\{\rho_r(b, a) : (b, a) \in \Omega\}, K\big),$$

and outputs

$$\mathbf{S}_{r,k} \leftarrow \mathbf{J}_{r,b^\star_{r,k},a^\star_{r,k}}, \qquad \boldsymbol{parent}_{r,k} \leftarrow b^\star_{r,k}, \qquad \boldsymbol{choice}_{r,k} \leftarrow a^\star_{r,k}.$$

Intuitively, at each level we expand every current beam by all $A$ levels, and keep the best $K$ expansions per row.

**GATHER$(\mathbf{Z}, \boldsymbol{parent})$.** This is the beam reindexing operator: it reorders a beam-dependent state tensor so that each surviving beam $k$ inherits the state of its selected parent beam $\boldsymbol{parent}_{r,k}$. Formally, if $\mathbf{Z} \in \mathbb{R}^{m\times K\times d}$, or more generally has any trailing shape $d$, then

$$\big(\textsc{Gather}(\mathbf{Z}, \boldsymbol{parent})\big)_{r,k,:} = \mathbf{Z}_{r,\boldsymbol{parent}_{r,k},:}, \qquad \forall r \in [m],\, k \in [K].$$

We apply this to $\mathbf{Q}_{\mathrm{blk}}$, $\mathbf{T}$, and the ancestry map $\boldsymbol{\pi}$, so that their beam dimensions remain consistent with the pruned score tensor $\mathbf{S}$.

**PICKLEVEL$(\mathbf{V}, \boldsymbol{choice})$.** Given candidate quantization values $\mathbf{V} \in \mathbb{R}^{m\times A}$ for the current column and selected level indices $\boldsymbol{choice} \in \mathbb{R}^{m\times K}$, PICKLEVEL returns the chosen quantized values per row and beam:

$$\big(\textsc{PickLevel}(\mathbf{V}, \boldsymbol{choice})\big)_{r,k} = \mathbf{V}_{r,\boldsymbol{choice}_{r,k}}, \qquad \forall r \in [m],\, k \in [K].$$

This is exactly the column update $\mathbf{Q}_{\mathrm{blk}}[:, :, j]$ for the surviving beams.

In implementation, TOPK corresponds to a row-wise `topk` on the flattened beam–level axis, while GATHER and PICKLEVEL are indexed gathers along the beam and level dimensions, respectively.

Table 5: Dimensions of major tensors used in CoreQ and $K$-CoreQ rounding for one linear layer of shape $m \times n$. $B$ is the block size, $K$ is the beam width, and $A$ is the number of quantization levels. The center correction $\mathbf{T}$ used in lazy-batch update is defined in Algorithms 2 and 3.

| Tensor | CoreQ ($K$=1) | $K$-CoreQ (beam) |
|---|---|---|
| Corrected rounding center $\mathbf{W}_{\mathrm{corr}}$ | $m \times n$ | $m \times n$ |
| Cholesky factor $\mathbf{L}$ | $n \times n$ | $n \times n$ |
| Quantized weight $\mathbf{Q}$ | $m \times n$ | $m \times n$ |
| Beam scores $\mathbf{S}$ | — | $m \times K$ |
| Suffix decisions $\mathbf{Q}_{\mathrm{tail}}$ | — | up to $m \times K \times n$ |
| Per-step expansion cost $\boldsymbol{\Delta}$ | — | $m \times K \times A$ |
| Beam ancestry $\boldsymbol{\pi}$ | — | $m \times K$ |
| Block decisions $\mathbf{Q}_{\mathrm{blk}}$ | $m \times B$ | $m \times K \times B$ |
| Center correction $\mathbf{T}$ | $m \times B$ | $m \times K \times B$ |
| Interference-cancelled center $\widetilde{\mathbf{W}}$ | $m \times B$ | $m \times K$ |

**Algorithm 1:** $\alpha_{\text{corr}}$: closed-form correction coefficient

**Input:** FP weight $\mathbf{W} \in \mathbb{R}^{m \times n}$, teacher input $\mathbf{X}_f \in \mathbb{R}^{n \times N}$, student input $\mathbf{X}_q \in \mathbb{R}^{n \times N}$, damping $\lambda \geq 0$

**Output:** Correction coefficient $\alpha_{\text{corr}} \in [0, 1]$, shifted target $\mathbf{W}_{\text{corr}} \in \mathbb{R}^{m \times n}$

$\mathbf{H} \leftarrow \mathbf{X}_q \mathbf{X}_q^\top + \lambda \mathbf{I}$ // Hessian with damping

$\mathbf{U} \leftarrow \mathbf{W}(\mathbf{X}_f - \mathbf{X}_q)$ // layerwise mismatch

$\widetilde{\mathbf{S}} \leftarrow \mathbf{U} \mathbf{X}_q^\top \mathbf{H}^{-1}$ // empirical reachable correction

$\mathbf{s} \leftarrow \widetilde{\mathbf{S}} \mathbf{X}_q$ // reachable component

$\eta \leftarrow \mathbf{U} - \mathbf{s}$ // residual

$\alpha_{\text{corr}} \leftarrow \dfrac{\|\mathbf{s}\|_F^2}{\|\mathbf{s}\|_F^2 + \|\eta\|_F^2}$

$\mathbf{W}_{\text{corr}} \leftarrow \mathbf{W} + \alpha_{\text{corr}} \widetilde{\mathbf{S}}$ // shifted rounding center

**return** $(\alpha_{\text{corr}},\ \mathbf{W}_{\text{corr}})$

**Algorithm 2:** CoreQ: greedy successive rounding with lazy-batch update

**Input:** FP weight $\mathbf{W} \in \mathbb{R}^{m \times n}$, teacher input $\mathbf{X}_f$, student input $\mathbf{X}_q$, permutation $\mathbf{P}$, block size $B$, damping $\lambda \geq 0$

**Output:** Quantized weight $\mathbf{Q}$

$(\alpha_{\text{corr}},\ \mathbf{W}_{\text{corr}}) \leftarrow \textsc{ComputeAlpha}(\mathbf{W}, \mathbf{X}_f, \mathbf{X}_q, \lambda)$ // Algorithm 1

$\mathbf{H} \leftarrow \mathbf{P}^\top (\mathbf{X}_q \mathbf{X}_q^\top + \lambda \mathbf{I}) \mathbf{P}$

$\mathbf{L} \leftarrow \textsc{Cholesky}(\mathbf{H})$

$\mathbf{L} \leftarrow \mathbf{L} / \mathrm{diag}(\mathbf{L}) - \mathbf{I}$

$\mathbf{W}_{\text{corr}} \leftarrow \mathbf{W}_{\text{corr}} \mathbf{P}$ // permute target

$\mathbf{Q} \in \mathbb{R}^{m \times n} \leftarrow \mathbf{0}$

**for** $i = n,\ n - B,\ n - 2B, \ldots$ **do**

  $\mathbf{T} \leftarrow (\mathbf{W}_{\text{corr}}[:, i:] - \mathbf{Q}[:, i:])\, \mathbf{L}[i:,\ i{-}B : i] \in \mathbb{R}^{m \times B}$ // tail correction

  $\widetilde{\mathbf{W}} \in \mathbb{R}^{m \times B} \leftarrow \mathbf{0}$

  **for** $j = B - 1, B - 2, \ldots, 0$ **do**

    $t \leftarrow i - B + j$ // absolute column index

    $\widetilde{\mathbf{W}}[:, j] \leftarrow \mathbf{W}_{\text{corr}}[:, t] + (\mathbf{W}_{\text{corr}}[:, i{-}B : i] - \mathbf{Q}[:, i{-}B : i])\, \mathbf{L}[i{-}B : i,\ t] + \mathbf{T}[:, j]$

    $\mathbf{Q}[:, t] \leftarrow \textsc{Round}(\widetilde{\mathbf{W}}[:, j])$

  **end**

**end**

$\mathbf{Q} \leftarrow \mathbf{Q} \mathbf{P}^\top$ // undo permutation

**return** $\mathbf{Q}$

**Algorithm 3:** $K$-CoreQ: beam-search rounding with lazy-batch update

**Input:** FP weight $\mathbf{W} \in \mathbb{R}^{m\times n}$, teacher input $\mathbf{X}_f$, student input $\mathbf{X}_q$, permutation $\mathbf{P}$, block size $B$, beam width $K$, damping $\lambda \geq 0$

**Output:** Quantized weight $\mathbf{Q}$

$(\alpha_{\text{corr}},\, \mathbf{W}_{\text{corr}}) \leftarrow \textsc{ComputeAlpha}(\mathbf{W}, \mathbf{X}_f, \mathbf{X}_q, \lambda)$ `// Algorithm 1`

$\mathbf{H} \leftarrow \mathbf{P}^\top(\mathbf{X}_q\mathbf{X}_q^\top + \lambda\mathbf{I})\mathbf{P}$

$\mathbf{L} \leftarrow \textsc{Cholesky}(\mathbf{H})$

$\mathbf{L} \leftarrow \mathbf{L}/\mathrm{diag}(\mathbf{L}) - \mathbf{I}$

$\mathbf{W}_{\text{corr}} \leftarrow \mathbf{W}_{\text{corr}}\mathbf{P}$ `// permute target`

**Initialize beam:**

$\mathbf{S} \in \mathbb{R}^{m\times K} \leftarrow +\infty,\ \mathbf{S}_{:,0} \leftarrow 0$ `// beam scores (per row)`

$\mathbf{Q}_{\text{tail}} \in \mathbb{R}^{m\times K\times 0} \leftarrow \emptyset$ `// accumulated suffix decisions`

**for** $i = n,\ n-B,\ n-2B, \ldots$ **do**

  $\mathbf{T} \leftarrow (\mathbf{W}_{\text{corr}}[:, i:] - \mathbf{Q}_{\text{tail}})\,\mathbf{L}[i:,\, i-B:i] \in \mathbb{R}^{m\times K\times B}$ `// tail correction`

  $\mathbf{Q}_{\text{blk}} \in \mathbb{R}^{m\times K\times B} \leftarrow \mathbf{0}$

  $\boldsymbol{\pi} \in \mathbb{R}^{m\times K} \leftarrow [0, 1, \ldots, K-1]$ `// beam ancestry`

  **for** $j = B-1, B-2, \ldots, 0$ **do**

    $t \leftarrow i - B + j$ `// absolute column index`

    $\widetilde{\mathbf{W}} \leftarrow \mathbf{W}_{\text{corr}}[:,t] + (\mathbf{W}_{\text{corr}}[:, i-B:i] - \mathbf{Q}_{\text{blk}})\,\mathbf{L}[i-B:i,\, t] + \mathbf{T}[:,:,j] \in \mathbb{R}^{m\times K}$

    $\mathbf{V} \leftarrow \textsc{Levels}(\widetilde{\mathbf{W}}) \in \mathbb{R}^{m\times A}$ `// candidate quantization levels`

    $\boldsymbol{\Delta} \leftarrow \mathbf{L}[t,t]^2\,|\widetilde{\mathbf{W}} - \mathbf{V}|^2 \in \mathbb{R}^{m\times K\times A}$

    $(\mathbf{S},\, \boldsymbol{parent},\, \boldsymbol{choice}) \leftarrow \textsc{TopK}(\mathbf{S} + \boldsymbol{\Delta},\ K)$ `// keep best` $K$

    $\mathbf{Q}_{\text{blk}} \leftarrow \textsc{Gather}(\mathbf{Q}_{\text{blk}}, \boldsymbol{parent})$

    $\mathbf{T} \leftarrow \textsc{Gather}(\mathbf{T}, \boldsymbol{parent})$

    $\boldsymbol{\pi} \leftarrow \textsc{Gather}(\boldsymbol{\pi}, \boldsymbol{parent})$

    $\mathbf{Q}_{\text{blk}}[:,:,j] \leftarrow \textsc{PickLevel}(\mathbf{V}, \boldsymbol{choice})$ `// selected level for column` $t$

  **end**

  $\mathbf{Q}_{\text{tail}} \leftarrow \textsc{Concat}\big(\mathbf{Q}_{\text{blk}},\ \textsc{Gather}(\mathbf{Q}_{\text{tail}}, \boldsymbol{\pi})\big)$

**end**

$k_r^\star \leftarrow \arg\min_{k\in[K]} \mathbf{S}_{r,k},\ \ \forall r \in [m]$ `// best beam per row`

$\mathbf{Q}_p[r,:] \leftarrow \mathbf{Q}_{\text{tail}}[r,\, k_r^\star,\, :],\ \ \forall r \in [m]$

$\mathbf{Q} \leftarrow \mathbf{Q}_p\mathbf{P}^\top$ `// undo permutation`

**return** $\mathbf{Q}$

# C Additional Results

## C.1 Analysis of the correction coefficient $\alpha$

**Why a single $\alpha$ does not suffice.** A natural hypothesis is that increasing $\alpha$, which pushes the calibration target toward the full-precision activation, should monotonically reduce the layerwise mean activation error (MAE) $\|\mathbf{X}_{\text{f}}^{\ell} - \mathbf{X}_{\text{q}}^{\ell}\|$. Figure 5 (top row) confirms this on the calibration set: across all three models, larger $\alpha$ yields smaller MAE, with the effect concentrated in the deeper layers where upstream errors have accumulated. The trend does not transfer to held-out data. On the validation set (bottom row), $\alpha = 0.5$ matches or improves on $\alpha = 1$ at every layer for L2-7B and L3-8B, and the gap widens with depth. On L2-13B, although $\alpha = 1$ achieves the lowest calibration MAE in the deepest layers, it diverges from $\alpha = 0.5$ on the validation set, and the gap widens with depth. This shows that aggressive correction overfits to calibration-specific mismatch patterns rather than recovering population-level structure. Yet a constant choice is itself fragile. The validation MAE curves diverge across $\alpha$ in a layer-dependent fashion within each model (Figure 5), so even the best per-model constant is suboptimal at different depths.

**Empirical behavior of $\alpha_{\text{corr}}$.** The closed-form $\alpha_{\text{corr}}$ is computed per layer from the calibration batch alone, with no held-out data or grid search. Figure 2 reports its values across four models from LLaMA-2-7B to LLaMA-2-70B at 3-bit per-channel quantization. Two structural properties emerge.

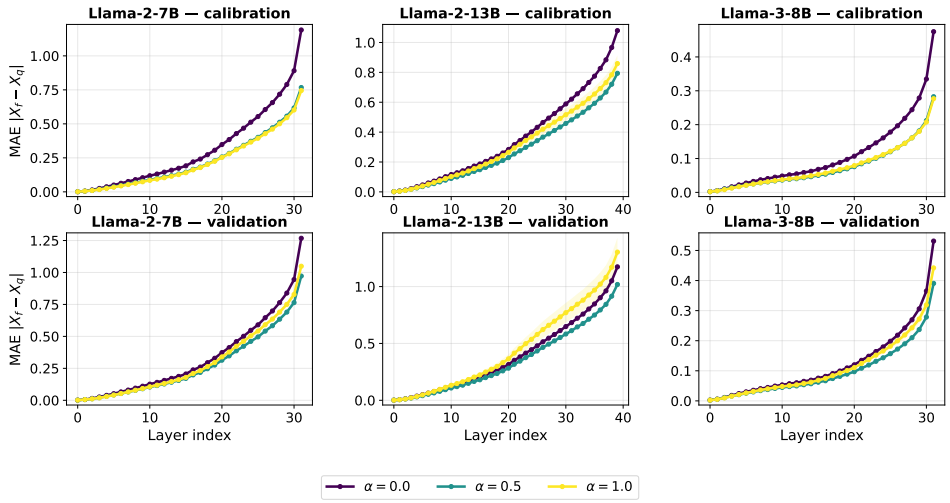



Figure 5: Mean activation error (MAE) measured on (top) calibration and (bottom) validation set.

**(i)** $\alpha_{\text{corr}}$ **varies systematically across layer types**: for example, the output projection $\mathbf{W}_O$ consistently requires the most aggressive correction ($\alpha_{\text{corr}} \approx 0.4$–$0.6$ in steady state) and the ordering persists across model family and an order-of-magnitude scale change. One plausible explanation is that $\mathbf{W}_O$, as the final linear map in the attention branch, sees mismatch after attention aggregation and head mixing, which often appears as a coherent feature-space shift that can be absorbed by a linear output projection. In contrast, $\mathbf{W}_V$ is applied before the data-dependent attention mixing; its mismatch is more token- and context-specific and therefore less well represented by a single linear correction on $\mathbf{X}_{\text{q}}$. The fact that this ordering persists across model families and scales suggests that the coefficient captures layer-specific geometry of the mismatch, rather than random calibration noise.

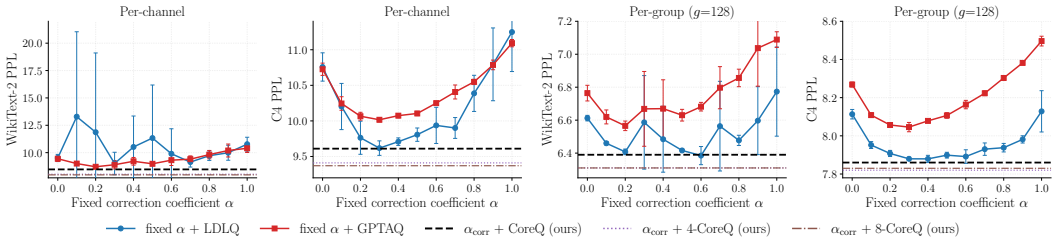
**(ii)** $\alpha_{\text{corr}}$ **varies systematically across depth.** Early blocks transiently approach $\alpha_{\text{corr}} \approx 1$ before settling to layer-specific steady points, reflecting the accumulation of upstream rounding error layer by layer. This is consistent with the reachability interpretation: near the beginning of the network, the activation mismatch is induced by only a short quantized prefix and is often close to a coherent first-order perturbation, making it largely explainable by a local linear weight shift. As depth increases, upstream rounding errors pass through attention, MLP, normalization, and residual connections, making the mismatch more mixed and layer-dependent. The reachable fraction therefore stabilizes to layer-specific levels rather than remaining uniformly close to one. This behavior is desirable in PTQ: once propagated mismatch becomes less locally reachable, full correction would allow accumulated calibration-specific error to steer the calibration target. The depth-adaptive shrinkage instead corrects only the component that remains explainable by the current layer and otherwise falls back toward the standard calibration target.

### C.2 Ablation with Correction Strength and Beam Width

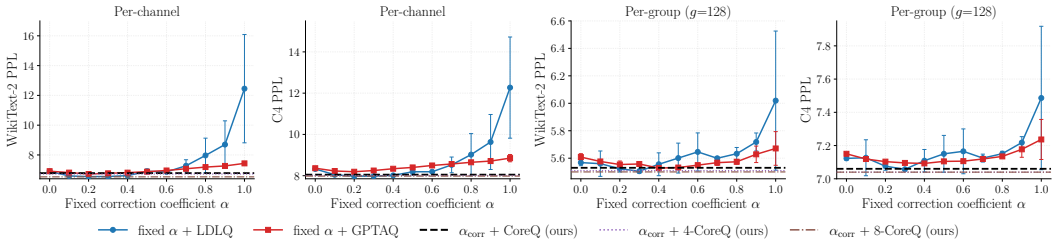
We jointly vary the correction coefficient and beam width in a GPTQ-based rounding implementation to examine their respective contributions (Table 6). Fixed coefficients are shared across layers, whereas $\alpha_{\text{corr}}$ is computed separately for each layer. At $K = 1$, replacing the standard target ($\alpha = 0$) with the adaptive target reduces WikiText-2 perplexity from $9.36$ to $8.41$. Increasing the beam width alone to $K = 8$ yields $9.03$, while combining adaptive correction with $K = 8$ yields $7.97$. These results demonstrate the complementary benefits of adaptive target selection and expanded search.

### C.3 Runtime Breakdown

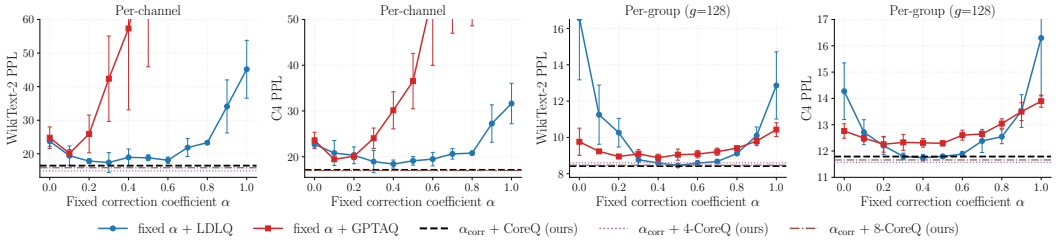
The quantization times in the main tables are end-to-end wall-clock measurements and include model execution, calibration-data handling, data movement, and non-GPU overhead. To isolate the computational stages of CoreQ, Table 7 instead reports CUDA-event GPU time for LLaMA-2-13B. The collection of $\mathbf{X}_{\text{q}}$, construction of $\mathbf{H}$, and greedy rounding are shared with the corresponding LDLQ pipeline. CoreQ additionally collects $\mathbf{X}_{\text{f}}$ and forms $\mathbf{C} = \mathbf{X}_{\text{f}}\mathbf{X}_{\text{q}}^{\top}$, which account for most of




(a) LLaMA-2-7B




(b) LLaMA-2-13B




(c) LLaMA-3-8B

Figure 6: Fixed-$\alpha$ sweep on 3-bit weight-only quantization. Each panel reports perplexity on WikiText-2 or C4 under per-channel and per-group ($g$=128) granularity, mean $\pm$ std over 5 calibration seeds. Solid lines sweep fixed $\alpha$ for LDLQ and GPTAQ; horizontal bands show CoreQ and its bounded-search variants $K$-CoreQ ($K \in \{4, 8\}$) using the closed-form $\alpha_{\text{corr}}$.

Table 6: Ablation of correction coefficient $\alpha$ and beam width $K$ using GPTQ-based rounding on LLaMA-2-7B with 3-bit per-channel weight-only quantization. Fixed-$\alpha$ settings use a global coefficient, whereas CoreQ uses the layer-wise coefficient $\alpha_{\text{corr}}$. We report mean $\pm$ standard deviation of perplexity (lower is better). The lowest mean in each column is bold.

| | WikiText-2 | | | C4 | | |
|---|---|---|---|---|---|---|
| Correction | $K=1$ | $K=4$ | $K=8$ | $K=1$ | $K=4$ | $K=8$ |
| $\alpha = 0.00$ | $9.36 \pm 0.24$ | $9.09 \pm 0.11$ | $9.03 \pm 0.10$ | $10.71 \pm 0.09$ | $10.48 \pm 0.16$ | $10.39 \pm 0.02$ |
| $\alpha = 0.25$ | $8.91 \pm 0.25$ | $8.57 \pm 0.04$ | $8.51 \pm 0.18$ | $10.05 \pm 0.08$ | $9.93 \pm 0.05$ | $9.85 \pm 0.02$ |
| $\alpha = 0.50$ | $8.94 \pm 0.17$ | $8.67 \pm 0.12$ | $8.83 \pm 0.35$ | $10.11 \pm 0.03$ | $10.02 \pm 0.03$ | $9.90 \pm 0.01$ |
| $\alpha = 0.75$ | $9.80 \pm 0.42$ | $9.19 \pm 0.04$ | $9.14 \pm 0.27$ | $10.51 \pm 0.12$ | $10.25 \pm 0.03$ | $10.19 \pm 0.03$ |
| $\alpha = 1.00$ | $10.40 \pm 0.38$ | $9.90 \pm 0.17$ | $10.37 \pm 0.44$ | $11.09 \pm 0.05$ | $10.86 \pm 0.04$ | $10.76 \pm 0.05$ |
| CoreQ ($\alpha_{\text{corr}}$) | $\mathbf{8.41} \pm 0.21$ | $\mathbf{8.03} \pm 0.14$ | $\mathbf{7.97} \pm 0.07$ | $\mathbf{9.61} \pm 0.12$ | $\mathbf{9.41} \pm 0.05$ | $\mathbf{9.37} \pm 0.01$ |

its additional GPU cost. These computations are also required by mismatch-aware baselines such as GPTAQ. In comparison, evaluating the closed-form coefficient accounts for only $1.1\%$ of the profiled GPU time.

### C.4 Sensitivity to Calibration Data Size

We evaluate the sensitivity of CoreQ to the number of calibration samples used to estimate the activation statistics and mismatch correction. Table 8 reports results for LLaMA-2-7B under 3-bit per-

Table 7: CUDA-event GPU-time profile of CoreQ for LLaMA-2-13B. Percentages are relative to the sum of the five profiled stages and exclude non-GPU wall-clock overhead.

| Stage | Time (s) | Share (%) |
|---|---|---|
| $\mathbf{X}_{\mathrm{q}}$ collection and $\mathbf{H} = \mathbf{X}_{\mathrm{q}}\mathbf{X}_{\mathrm{q}}^{\top}$ | 193.2 | 28.8 |
| $\mathbf{X}_{\mathrm{f}}$ collection | 23.1 | 3.4 |
| $\mathbf{C} = \mathbf{X}_{\mathrm{f}}\mathbf{X}_{\mathrm{q}}^{\top}$ | 122.1 | 18.2 |
| $\alpha_{\mathrm{corr}}$ computation | 7.7 | 1.1 |
| Greedy successive rounding | 324.5 | 48.4 |
| Total profiled GPU time | 670.6 | 100.0 |

Table 8: Sensitivity to calibration set size. LLaMA-2-7B with 3-bit per-channel weight-only quantization, calibrated on C4 (sequence length 2048). Perplexity on Wikitext-2 (Wiki2) and C4 (lower is better), and average zero-shot accuracy across six commonsense reasoning tasks (higher is better).

| | Wiki2 PPL (↓) | | C4 PPL (↓) | | Avg. Acc. (%, ↑) | |
|---|---|---|---|---|---|---|
| # Samples | GPTAQ | CoreQ | GPTAQ | CoreQ | GPTAQ | CoreQ |
| 16 | $11.09 \pm 0.34$ | $11.03 \pm 1.00$ | $12.13 \pm 0.19$ | $11.34 \pm 0.14$ | $58.17 \pm 0.48$ | $59.28 \pm 0.61$ |
| 32 | $10.37 \pm 1.35$ | $9.17 \pm 0.31$ | $10.80 \pm 0.10$ | $10.06 \pm 0.07$ | $60.10 \pm 0.23$ | $60.96 \pm 0.70$ |
| 64 | $9.28 \pm 0.57$ | $9.06 \pm 0.65$ | $10.26 \pm 0.02$ | $9.82 \pm 0.22$ | $60.86 \pm 0.64$ | $61.01 \pm 0.80$ |
| 128 | $8.84 \pm 0.23$ | $8.41 \pm 0.21$ | $10.04 \pm 0.06$ | $9.61 \pm 0.12$ | $60.91 \pm 0.52$ | $61.46 \pm 0.50$ |

channel weight-only quantization, using C4 calibration sets of different sizes. Across all calibration sizes, CoreQ achieves lower mean perplexity and higher average zero-shot accuracy than GPTAQ. The improvement is especially clear for C4 perplexity, where CoreQ consistently reduces error even with only 16 or 32 calibration samples. This suggests that the reachable/unreachable decomposition provides a useful safeguard against fully applying the mismatch when calibration data are limited. As the calibration set grows, both methods improve, reflecting more stable activation statistics and mismatch estimates. CoreQ continues to provide additional gains at 128 samples, indicating that the adaptive correction does not merely help in extremely data-scarce regimes.

### C.5 Additional Quantization Settings

**Joint weight-activation-KV cache quantization** To examine the smaller gains under joint quantization, we analyze QuaRot-rotated LLaMA-2-7B with 4-bit weights under W4A16KV16, W4A4KV16, and W4A4KV4. Table 9 reports the mean and standard deviation of $\alpha_{\mathrm{corr}}$ across 32 layers. Its mean decreases from $0.218$ under W4A16KV16 to $0.191$ under W4A4KV16 and $0.185$ under W4A4KV4. Thus, as activations and KV caches are quantized, a smaller fraction of the observed mismatch is reachable through current-layer weight refinement. This trend is consistent with the smaller performance gains of CoreQ under joint quantization.

Table 9: Correction geometry and CoreQ perplexity for QuaRot-rotated LLaMA-2-7B with 4-bit weights. Values of $\alpha_{\mathrm{corr}}$ are mean $\pm$ standard deviation across 32 layers.

| Metric | W4A16KV16 | W4A4KV16 | W4A4KV4 |
|---|---|---|---|
| $\alpha_{\mathrm{corr}}$ | $0.218 \pm 0.180$ | $0.191 \pm 0.151$ | $0.185 \pm 0.132$ |
| WikiText-2 / C4 PPL | 5.63 / 7.10 | 5.82 / 7.54 | 5.94 / 7.69 |

**Weight-only quantization with outlier reduction** Table 10 compares CoreQ against two outlier-aware weight-only baselines: AWQ [36], which migrates outliers from activations into weights via per-channel scaling, and OmniQuant [37], which learns quantization parameters via gradient descent. Combined with QuaRot [38] to handle activation outliers, CoreQ achieves the lowest Wiki2 perplexity, for both per-channel and per-group quantization across all three models, while running 3–5$\times$ faster than OmniQuant. It also improves C4 perplexity over OmniQuant in every setting. AWQ achieves lower C4 perplexity on LLaMA-2-7B and LLaMA-2-13B under per-group quantization, but requires longer quantization time than QuaRot+CoreQ for both models. On LLaMA-3-8B with per-channel quantization, QuaRot+CoreQ reduces WikiText-2 perplexity from $12.18$ for AWQ and

15.11 for OmniQuant to 7.14. These results show that CoreQ composes well with outlier-reduction preprocessing and achieves strong accuracy–cost trade-offs in this setting.

Table 10: W3 weight-only quantization: perplexity on Wiki2/C4 and wall-clock quantization time (seconds, single GPU). Perplexities are mean ± std over calibration seeds; quantization time is mean over the same runs.

| | LLaMA-2-7B | | | LLaMA-2-13B | | | LLaMA-3-8B | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | Wiki2 | C4 | Time | Wiki2 | C4 | Time | Wiki2 | C4 | Time |
| *Per-channel (W3)* | | | | | | | | | |
| AWQ | 16.93 ± 0.33 | 18.55 ± 0.45 | 801 | 6.48 ± 0.02 | 7.92 ± 0.02 | 1326 | 12.18 ± 0.16 | 16.48 ± 0.20 | 777 |
| OmniQuant | 6.63 ± 0.01 | 8.64 ± 0.02 | 2309 | 5.60 ± 0.02 | 7.46 ± 0.03 | 3975 | 15.11 ± 0.47 | 21.10 ± 0.74 | 2338 |
| QuaRot + CoreQ | **5.84 ± 0.01** | **7.70 ± 0.01** | **570** | **5.17 ± 0.00** | **6.99 ± 0.00** | **956** | **7.14 ± 0.01** | **11.09 ± 0.04** | **679** |
| *Group-wise, $g$=128 (W3)* | | | | | | | | | |
| AWQ | 6.19 ± 0.01 | **7.33 ± 0.01** | 843 | 5.31 ± 0.01 | **6.57 ± 0.01** | 1419 | 8.18 ± 0.02 | 11.11 ± 0.01 | **838** |
| OmniQuant | 6.10 ± 0.01 | 7.81 ± 0.01 | 2334 | 5.32 ± 0.00 | 7.02 ± 0.00 | 4147 | 8.81 ± 0.04 | 12.50 ± 0.02 | 2438 |
| QuaRot + CoreQ | **5.78 ± 0.00** | 7.58 ± 0.00 | **787** | **5.12 ± 0.00** | 6.91 ± 0.00 | **1282** | **6.99 ± 0.01** | **10.74 ± 0.04** | 903 |

**Vision transformers** To verify that CoreQ's calibration objective transfers beyond language models, we evaluate W4A4 quantization on DeiT-S and DeiT-B [48] calibrated with 128 ImageNet samples (Table 11). CoreQ outperforms both GPTQ and GPTAQ at comparable wall-clock cost: on DeiT-S it improves top-1 accuracy by 3.47 over GPTQ and 1.32 over GPTAQ, and on DeiT-B by 0.95 and 0.37 respectively. The bounded-search variants (4-CoreQ and 8-CoreQ) provide further gains on DeiT-S at modest extra cost; on DeiT-B the marginal benefit saturates at $K = 8$. These results indicate that CoreQ is not specific to causal language modeling, and it captures structure of the layerwise mismatch that arises in transformer architectures generally.

Table 11: W4A4 quantization results of vision transformers, calibrated using 128 samples of ImageNet. All results are averaged over 5 random seeds.

| Method | DeiT-S | | DeiT-B | |
|---|---|---|---|---|
| | Acc (%) | Q.Time (s) | Acc (%) | Q.Time (s) |
| FP16 | 79.9 | – | 81.8 | – |
| GPTQ | 71.94 ± 0.06 | 13.86 ± 2.09 | 77.60 ± 0.04 | 18.03 ± 0.30 |
| GPTAQ | 74.09 ± 0.16 | 15.40 ± 0.70 | 78.18 ± 0.12 | 24.85 ± 0.75 |
| CoreQ | 75.41 ± 0.07 | 14.75 ± 0.06 | 78.55 ± 0.09 | 21.20 ± 0.70 |
| 4-CoreQ | **75.52** ± 0.08 | 17.41 ± 0.31 | 78.56 ± 0.12 | 24.74 ± 1.87 |
| 8-CoreQ | **75.51** ± 0.07 | 17.62 ± 0.41 | **78.67** ± 0.10 | 25.18 ± 0.51 |

Table 12: W3 weight-only quantization on Qwen3-8B and Phi-3-3.8B. We report perplexity on WikiText-2 and C4 (↓), average zero-shot accuracy across six commonsense reasoning tasks (↑), and wall-clock quantization time (seconds). Mean ± std over 5 calibration seeds. Best per (granularity, model, metric) in bold.

| | | Qwen3-8B | | | | Phi-3-3.8B | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Granularity** | **Method** | **Wiki2** (↓) | **C4** (↓) | **Avg. Acc** (↑) | **Q.Time (s)** | **Wiki2** (↓) | **C4** (↓) | **Avg. Acc** (↑) | **Q.Time (s)** |
| | FP16 | 9.73 | 14.65 | 74.15 | — | 6.01 | 9.11 | 75.41 | — |
| Per-channel | GPTQ | 18.75 ± 0.45 | 19.69 ± 0.12 | 54.17 ± 0.51 | 499.2 ± 17.7 | 15.46 ± 0.50 | 15.80 ± 0.05 | 54.48 ± 1.48 | 266.5 ± 7.6 |
| | GPTAQ | 17.63 ± 0.59 | 19.08 ± 0.28 | 57.07 ± 0.90 | 647.2 ± 19.1 | 12.99 ± 0.28 | 14.21 ± 0.15 | 57.19 ± 0.85 | 335.7 ± 7.7 |
| | LDLQ | 16.18 ± 0.18 | 18.49 ± 0.08 | 56.19 ± 0.81 | 375.4 ± 13.4 | 15.75 ± 0.74 | 16.03 ± 0.48 | 54.49 ± 0.65 | 216.1 ± 2.8 |
| | CoreQ | **15.44 ± 0.23** | **17.51 ± 0.08** | **63.58 ± 0.93** | 553.9 ± 12.0 | **11.86 ± 0.34** | **13.68 ± 0.32** | **60.51 ± 1.30** | 302.1 ± 5.8 |
| Per-group ($g$128) | GPTQ | 11.75 ± 0.25 | 15.09 ± 0.11 | 68.49 ± 0.51 | 494.9 ± 11.0 | 8.84 ± 0.07 | 11.31 ± 0.07 | 67.84 ± 0.48 | 262.7 ± 4.6 |
| | GPTAQ | 11.95 ± 0.08 | 15.13 ± 0.06 | 68.94 ± 0.42 | 670.4 ± 9.7 | **8.45 ± 0.02** | **10.97 ± 0.04** | **68.74 ± 0.29** | 332.0 ± 4.0 |
| | LDLQ | **11.41 ± 0.05** | **14.65 ± 0.06** | 69.26 ± 0.78 | 419.5 ± 10.9 | 9.07 ± 0.29 | 11.46 ± 0.25 | 67.01 ± 0.92 | 231.9 ± 6.3 |
| | CoreQ | 11.50 ± 0.09 | 14.76 ± 0.04 | **70.49 ± 0.93** | 566.4 ± 14.8 | **8.40 ± 0.34** | **10.90 ± 0.27** | 67.95 ± 2.15 | 308.8 ± 10.3 |

Table 13: Symmetric 2-bit and 3-bit weight-only quantization on LLaMA-2-70B. We report perplexity on WikiText-2 and C4 (↓) and wall-clock quantization time (seconds) under per-channel and per-group ($g$=128) granularity. Mean ± std over 5 calibration seeds. Best per (bits, granularity, metric) in bold.

| | | **Per-channel** | | | **Per-group** ($g$=128) | | |
|---|---|---|---|---|---|---|---|
| **Bits** | **Method** | **Wiki2** (↓) | **C4** (↓) | **Q.Time (s)** | **Wiki2** (↓) | **C4** (↓) | **Q.Time (s)** |
| FP16 | — | 3.32 | 5.52 | — | 3.32 | 5.52 | — |
| W2 | GPTQ | 181.31 ± 19.81 | 83.17 ± 5.14 | 3301.2 ± 73.0 | 12.38 ± 1.42 | 11.08 ± 0.10 | 3271.6 ± 9.5 |
| | GPTAQ | 251.82 ± 80.72 | 83.59 ± 11.38 | 4663.4 ± 3.0 | 12.28 ± 3.07 | 11.85 ± 2.16 | 4846.5 ± 43.9 |
| | LDLQ | 193.51 ± 36.21 | 91.32 ± 10.08 | 2753.5 ± 26.7 | 8.68 ± 0.29 | 9.90 ± 0.22 | 2939.7 ± 76.5 |
| | CoreQ | **94.97 ± 12.49** | **38.46 ± 2.37** | 4731.9 ± 69.3 | **7.63 ± 0.49** | **8.95 ± 0.32** | 4772.1 ± 89.6 |
| W3 | GPTQ | 5.11 ± 0.02 | 6.77 ± 0.02 | 3256.0 ± 0.6 | 4.00 ± 0.01 | 5.95 ± 0.01 | 3323.5 ± 22.6 |
| | GPTAQ | 5.09 ± 0.05 | 6.73 ± 0.01 | 4727.5 ± 37.1 | 4.01 ± 0.02 | 5.95 ± 0.01 | 4731.5 ± 90.9 |
| | LDLQ | 5.06 ± 0.02 | 6.73 ± 0.02 | 2855.0 ± 137.9 | 3.97 ± 0.01 | 5.94 ± 0.00 | 2914.3 ± 0.9 |
| | CoreQ | **4.83 ± 0.03** | **6.52 ± 0.01** | 4785.1 ± 28.3 | **3.93 ± 0.01** | **5.90 ± 0.00** | 4783.5 ± 96.8 |

Table 14: Weight-only quantization of LLaMA-3.1-8B-Instruct on reasoning benchmarks. We report perplexity on WikiText-2 (↓), accuracy on MMLU (5-shot, ↑), and exact-match accuracy on GSM8K with chain-of-thought prompting (8-shot, ↑). Calibrated on 128 samples of C4. Mean ± std over 3 calibration seeds. Best result per (bits, granularity, metric) in bold.

| | | **Per-channel** | | | **Per-group** ($g$=128) | | |
|---|---|---|---|---|---|---|---|
| **Bits** | **Method** | **Wiki2** (↓) | **MMLU** (↑) | **GSM8K-CoT** (↑) | **Wiki2** (↓) | **MMLU** (↑) | **GSM8K-CoT** (↑) |
| FP16 | — | 6.24 | 67.95 | 75.66 | 6.24 | 67.95 | 75.66 |
| W3 | GPTQ | 154.03 ± 27.68 | 30.00 ± 3.07 | 0.28 ± 0.48 | 11.06 ± 0.48 | 56.33 ± 0.51 | 36.52 ± 1.38 |
| | GPTAQ | 33.79 ± 11.11 | 32.77 ± 2.31 | 2.86 ± 1.73 | 9.95 ± 0.22 | 56.20 ± 1.11 | 42.53 ± 3.21 |
| | LDLQ | 19.89 ± 1.53 | 39.62 ± 1.40 | 4.17 ± 0.92 | 9.72 ± 0.20 | 59.53 ± 0.62 | 45.69 ± 2.25 |
| | CoreQ | **18.75 ± 2.98** | **39.76 ± 2.83** | **8.62 ± 1.89** | **9.32 ± 0.14** | **59.95 ± 0.36** | **48.14 ± 1.67** |
| W4 | GPTQ | 188.91 ± 112.13 | 60.83 ± 2.28 | 38.19 ± 32.02 | 9.60 ± 1.03 | 65.67 ± 0.64 | 68.56 ± 0.88 |
| | GPTAQ | 13.21 ± 3.08 | 62.88 ± 0.34 | 58.98 ± 3.72 | 7.67 ± 0.02 | 65.98 ± 0.10 | 70.08 ± 0.09 |
| | LDLQ | 8.58 ± 0.12 | **64.35 ± 0.33** | 60.50 ± 1.20 | 7.64 ± 0.02 | 66.76 ± 0.19 | 69.87 ± 1.92 |
| | CoreQ | **8.22 ± 0.02** | 64.22 ± 0.20 | **63.83 ± 0.92** | **7.58 ± 0.02** | **66.91 ± 0.23** | **70.58 ± 0.80** |

# D Extended Discussion on Layer-wise Calibration Objectives for Compression

A widely used family of compression methods minimizes a quadratic reconstruction error that measures the layer's output distortion under a second-order (Hessian) surrogate [7, 8, 9, 10], with GPTQ being the widely adopted LLM-scale realization [11]. These methods share a common assumption: each layer's input activation is treated as fixed, and only the layer-local reconstruction error is optimized. Several recent works incorporate information about end-to-end model behavior into this proxy. OAC introduces an output-adaptive calibration objective motivated by output cross-entropy distortion and derives tractable approximations for layerwise optimization [40]. GuidedQuant integrates end-loss gradient information into the objective while modeling cross-weight dependencies within output channels [15]. YAQA targets end-to-end distribution preservation via Kronecker-factored approximations of the full-model KL Hessian and provides adaptive rounding guarantees [16]; and BoA exploits attention structure to derive a more informative local objective without backpropagation [17]. These methods improve calibration quality but typically incur substantially higher cost, as they require either end-loss gradient computation or more expensive Hessian approximations.

A complementary line of work addresses the observation that upstream compression introduces distributional shifts in layer activations. To mitigate this, several methods redefine the layer-wise calibration objective to account for activation mismatch. GPTAQ [12] targets the full mismatch objective—matching the full-precision layer output on clean activations against the quantized layer output on corrupted ones—and derives block-wise closed-form updates. ApiQ [14] adopts a related mismatch objective but embeds it within a learning-based pipeline to jointly optimize quantized weights and LoRA adapters. Closely related to our continuous formulation is QEP [13], which interpolates between standard Hessian-based calibration and full mismatch-aware calibration using a manually tuned, global propagation-strength coefficient. Alternatively, CBQ [18] relaxes strict per-layer independence, quantizing blocks of layers to absorb propagated errors into an expanded optimization unit. Finally, a structurally related motif appears in low-rank SVD compression, where SAES-SVD [49] balances accumulated upstream error against local distortion via an adaptive alignment coefficient.

**Addressing limitations** CoreQ differs from prior work in two respects. First, unlike GPTAQ, which assumes full mismatch correction is optimal, or QEP, which relies on a rigid global hyperparameter, CoreQ shrinks the correction with a closed-form per-layer coefficient motivated by the finite-calibration analysis in Theorem 3.1. Second, unlike CBQ or ApiQ, CoreQ requires neither multi-layer optimization units nor gradient-based learning. The corrected objective retains the standard Hessian-weighted quadratic form, so the bounded search of $K$-CoreQ applies directly with negligible overhead relative to standard local PTQ.

# E Broader Impacts

This work aims to improve the efficiency of large language model deployment through post-training quantization. By reducing memory footprint and inference cost, CoreQ may make LLMs more accessible on resource-constrained hardware and reduce the computational and energy cost of serving existing models. These benefits are especially relevant for deployment scenarios where full-precision inference is impractical. However, more efficient inference can also lower the barrier to misuse. Quantized models may be deployed more easily for spam generation, misinformation, automated manipulation, or other harmful applications. CoreQ does not introduce new model capabilities or alter the training data, but it can make existing capabilities cheaper to run. As a result, quantized models should be deployed with the same safety evaluations, access controls, and misuse-mitigation measures as their full-precision counterparts.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are stated in the abstract and introduction, and are supported by the method in Section 3 and experiments in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations and future directions are discussed in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The theoretical result is stated in Theorem 3.1, with proof and additional discussion in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4.1 describes the experimental setup, calibration data, evaluated models, baselines, metrics, seeds, and timing protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code repository is linked in the abstract; experiments use publicly available models and datasets described in Section 4.1.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 specifies the calibration setup, datasets, benchmarks, quantization settings, baselines, number of seeds, and hardware used for timing.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Main tables report mean and standard deviation over seeds, with the number of seeds stated in the captions and Section 4.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4.1 reports wall-clock quantization time on a single H200 GPU; additional time and memory results are reported in Section 4.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and the work uses publicly available models and datasets without human-subject experiments or private data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are discussed in Appendix E.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new pretrained model, dataset, or high-risk generative system.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the model and dataset creators in Section 4.1.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release public code for CoreQ through the repository linked in the abstract.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subject research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human-subject research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as part of the proposed method, experiments, or scientific analysis.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.