

Complementary Lattice Correction for Post-Training Quantization

Tung Nguyen^{1,2}, Quang Hung Tran¹, Linh Ngo Van¹, Hoang Thai Dinh²,
Diep Nguyen², Trung Le³, Duc Anh Nguyen^{1†},

¹Hanoi University of Science and Technology, Vietnam,

²University of Technology Sydney, Australia,

³Monash University, Australia

Abstract

Weight-only post-training quantization (PTQ) for large language models (LLMs) typically ends with the projection of floating-point weights onto a uniform integer lattice, introducing structured output error due to non-zero activation means. Existing bias correction methods compensate for this shift by adding bias terms, which is incompatible with modern bias-less architectures such as LLaMA and Mistral and only addresses the error indirectly. We propose **Complementary Lattice Correction (CLC)**, a lightweight post-quantization method that corrects projection error directly on the integer lattice without modifying the model architecture. CLC performs a budget-constrained greedy descent that selectively flips quantized weights to adjacent lattice levels. We further show that CLC provably reduces the first-moment error and admits a worst-case bound on the layer-wise mean-squared error change, which certifies non-increase under a verifiable budget condition. Experiments across multiple models, quantizers, and bit widths show that CLC consistently improves perplexity and downstream accuracy with negligible overhead.

1 Introduction

The deployment of large language models (LLMs) with tens or hundreds of billions of parameters is strictly constrained by memory bandwidth and storage capacity. Weight-only post-training quantization (PTQ) has become a dominant paradigm for relaxing these constraints, enabling low-bit inference with limited degradation (Frantar et al., 2023; Lin et al., 2024). Modern PTQ methods that quantize weights differ in how they shape the quantization problem, through activation-aware scaling (Xiao et al., 2023; Lin et al., 2024), learned rotations (Shao et al., 2024; Liu et al., 2025a; Sun et al., 2025), or second-order weight updates (Frantar et al., 2023; Nagel et al., 2020). Despite these

differences, they share a structural feature: the weight tensor is ultimately represented by discrete codes on an integer lattice, commonly obtained through nearest-level projection or an equivalent lattice assignment.

This projection introduces a quantization residual that does not fully average out under LLM activation statistics. Since LLM activations exhibit non-zero empirical means (Xiao et al., 2023), the residual induces a structured first-moment shift in the layer output, an effect that becomes more pronounced at low bit widths. Classical bias correction (Nagel et al., 2019) compensates for this drift by absorbing it into an additive output-side bias, while leaving the underlying weight quantization residual unchanged. This correction is well matched only to layers that already carry a native bias. Modern LLMs are predominantly composed of bias-free linear layers—for example LLaMA (Grattafiori et al., 2024) and Mistral (Jiang et al., 2023)—though some architectures retain biases in selected modules (e.g. certain attention projections in Qwen). For the bias-free majority, additive bias correction requires introducing new parameters and altering the deployment graph. CLC instead operates directly on the integer lattice and applies uniformly to both bias-containing and bias-free layers.

Figure 1 visualizes this phenomenon on a real LLaMA-3.1 (Grattafiori et al., 2024) layer. Round-to-nearest (RTN) quantization produces a structured signed output shift field rather than purely zero-mean noise, yielding a measurable drift across tokens and output channels. Motivated by this observation, we propose **Complementary Lattice Correction (CLC)**, a lightweight post-quantization method that operates directly in discrete weight space. Rather than compensating for quantization drift through auxiliary bias parameters, CLC operates directly in discrete weight space. Starting from an existing quantized representation, CLC performs targeted adjacent-level updates that selec-

[†]Corresponding author: anhnd@soict.hust.edu.vn

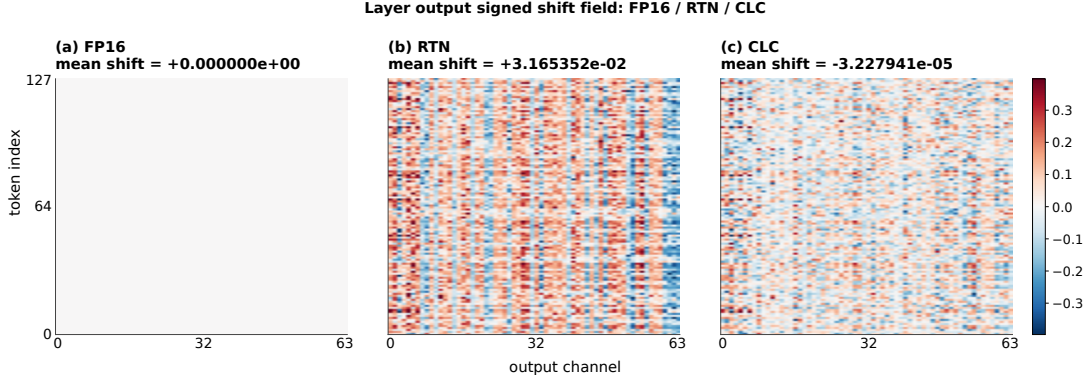


Figure 1: Visualization of the layer-wise signed output shift field $Y_q - Y$ on a real LLaMA-3.1 linear projection. Instead of behaving as zero-mean noise, the RTN residual induces a structured first-moment output shift across tokens and output channels, yielding a non-zero average output drift (mean shift = $+3.17 \times 10^{-2}$). CLC suppresses this structured drift directly on the integer lattice through adjacent-level corrections, reducing the mean shift to near zero (-3.23×10^{-5}) without introducing bias parameters.

tively reassign weights across neighboring lattice states, reducing the first-moment output shift induced by the projection while preserving the original model architecture and deployment graph.

Conceptually, CLC reframes post-quantization correction as a discrete lattice optimization problem. The method performs a budget-constrained greedy descent guided by activation statistics, enabling local discrete weight transitions that systematically reduce the structured first-moment drift induced by quantization. Importantly, the effect of CLC extends beyond mean-shift correction: we further show that the method reduces the total layer-wise output error while preserving the original quantized structure and inference graph.

We summarize our contributions as follows:

- We propose **Complementary Lattice Correction (CLC)**, a lightweight and architecture-preserving post-quantization method that reduces the first-moment output shift induced by quantization through budget-constrained adjacent-level updates.
- We provide theoretical analysis showing that CLC reduces the first-moment output error unconditionally, and bounds the total layer-wise MSE change, certifying non-increase under a verifiable budget condition.
- Experiments on recent open-source models demonstrate that CLC consistently improves low-bit quantization with negligible computational overhead.

2 Related Work

Quantization for efficient LLM inference.

Quantization reduces memory and compute by representing weights and/or activations with low-precision values. Existing approaches are commonly divided into quantization-aware training (QAT), which simulates low-precision computation during optimization (Jacob et al., 2018; Esser et al., 2020; Jung et al., 2019), and post-training quantization (PTQ), which compresses a pretrained model with little or no retraining (Yao et al., 2022; Dettmers et al., 2022; Xiao et al., 2023; Liu et al., 2025b). QAT can recover accuracy at very low precision, but typically requires access to the original optimization pipeline and substantial compute (Yao et al., 2021; Liu et al., 2024; Wang et al., 2025). PTQ is therefore attractive for LLM deployment because it can be applied directly to released checkpoints using only calibration data.

Post-training quantization. PTQ methods vary by quantization format, tensor scope, and calibration cost, including uniform versus non-uniform quantization, weight-only (Lee et al., 2024; Heo et al., 2024) versus weight-activation quantization (Liu et al., 2023; Wei et al., 2023; Zhao et al., 2024; Yao et al., 2023). Non-uniform methods adapt to LLM weight distributions through sparse/mixed representations (Dettmers et al., 2024; Kim et al., 2024), additive/vector codebooks (Egiazarian et al., 2024; Zhang et al., 2025), or lattice-based schemes (Tseng et al., 2024). Uniform weight quantization (Frantar et al., 2023; Lin et al., 2024) instead maps weights to an integer

grid with scales; the simplest baseline is round-to-nearest (RTN), which projects each weight to its nearest representable value. This work also mainly focuses on uniform quantization for model weights.

A large body of PTQ work reduces quantization error through local or block reconstruction objectives (Frantar and Alistarh, 2022; Li et al., 2021), adaptive rounding (Nagel et al., 2020), Hessian-aware weight updates and error compensation (Frantar et al., 2023), or layer-wise error propagation (Arai and Ichikawa, 2025). GPTQ (Frantar et al., 2023) is a canonical example of this direction, using Hessian-aware sequential updates to compensate quantization errors across remaining weights after rounding step. Another line of work calibrates the quantization geometry through transformations. AWQ (Lin et al., 2024) represents activation-aware channel scaling, a broader family of methods that protect salient weights or migrate activation difficulty into the weights (Xiao et al., 2023; Lin et al., 2024). Rotation-based methods reduce outlier structure before quantization (Ashkboos et al., 2024; Liu et al., 2025a; Shao et al., 2025), while FlatQuant (Sun et al., 2025) represents recent affine or flatness-oriented transformations that further smooth weights and activations (Sun et al., 2025; Shao et al., 2024; Lee et al., 2025). These methods differ in objective and calibration strategy, but they ultimately produce a discrete quantized weight representation. Classical bias correction (Nagel et al., 2019) instead absorbs the expected output shift into an additive bias term, which is less suitable for bias-less LLMs.

Knowledge distillation for model efficiency. Beyond low-bit quantization, knowledge distillation (KD) is a parallel route to efficient LLM inference, compressing a large teacher into a smaller student rather than lowering the precision of a fixed model (Vu et al., 2026; Nguyen et al., 2026; Vuong et al., 2026; Truong et al., 2025). KD reduces cost by changing *what* the student represents, whereas post-training quantization—and CLC in particular—operates on the discrete lattice of an already-fixed weight tensor. The two are thus complementary: a distilled student can still be quantized and corrected by CLC.

3 Method

This section develops CLC from theory to algorithm. We begin by analyzing the layer-wise quantization error under activation statistics, which

allows us to characterize what makes a discrete optimization procedure effective for reducing quantization-induced output error. We then instantiate the resulting conditions as a constructive algorithm. The analysis focuses primarily on weight quantization.

3.1 Setup

We analyze a single layer ℓ with full-precision weights $W \in \mathbb{R}^{d \times C}$, omitting layer superscripts when no confusion arises, where d is the input dimension and C is the number of output channels. For an activation $x \in \mathbb{R}^d$, channel j outputs $x^\top W_{:,j}$. Let $\mu := \mathbb{E}[x] \in \mathbb{R}^d$ and $\Sigma := \text{Cov}(x) \in \mathbb{R}^{d \times d}$ denote the population activation mean and covariance. In practice, μ is estimated from calibration activations $X = [x_1, \dots, x_m] \in \mathbb{R}^{d \times m}$ as $\hat{\mu} := m^{-1} \sum_{t=1}^m x_t$.

Let W_q be the quantized weights produced by the final projection of a base quantizer. For output channel j , let s_j be its lattice step, so that projected weights lie on $s_j \mathbb{Z}$. We define the quantization residual as $E := W_q - W$, $e_j := E_{:,j}$ with $|e_{i,j}| \leq s_j/2$ under nearest-level projection.

We measure layer error by the expected squared output difference between the quantized and full-precision layers:

$$\mathcal{R}_\ell(W_q) := \sum_{j=1}^C \mathbb{E} \left[(x^\top e_j)^2 \right]. \quad (1)$$

This is decomposed into:

$$\mathcal{R}_\ell(W_q) = \underbrace{\sum_{j=1}^C b_j^2}_{=: \mathcal{B}_\ell} + \underbrace{\sum_{j=1}^C e_j^\top \Sigma e_j}_{=: \mathcal{V}_\ell}, \quad (2)$$

where $b_j := \mu^\top e_j$ is the first-moment output shift of channel j , $\mathcal{B}_\ell$ is the squared mean-shift error, and $\mathcal{V}_\ell$ is the covariance-weighted reconstruction error (See detailed in Appendix A).

Most PTQ methods aim to reduce the layer-wise error $\mathcal{R}_\ell$. However, the full population activation statistics are rarely accessible; methods typically approximate them from a small calibration set. This gives the calibration mean-shift terms $\hat{b}_j := \hat{\mu}^\top e_j$, $\hat{\mathcal{B}}_\ell := \sum_{j=1}^C \hat{b}_j^2$. We use hats to indicate quantities computed with $\hat{\mu}$ in place of μ . CLC constructs corrected quantized weights $W'_q := W_q + \Delta$ while keeping every entry on its original lattice: $W_{q,i,j} + \Delta_{i,j} \in s_j \mathbb{Z}$, $\forall (i, j)$.

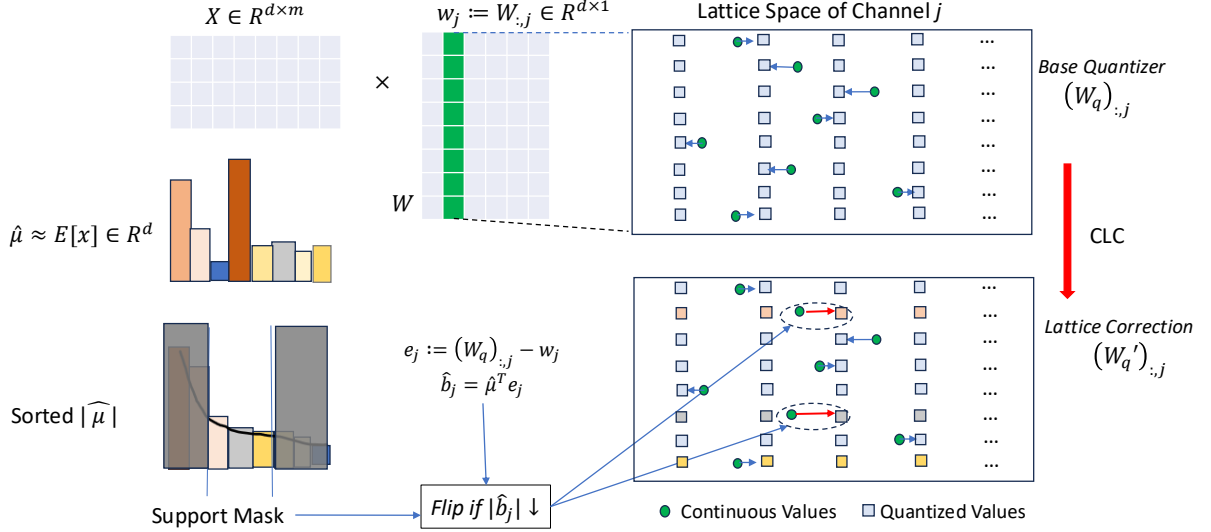


Figure 2: Illustration of Complementary Lattice Correction. Green dots are full-precision weights, and hollow squares are quantized lattice levels. Small blue arrows show RTN-guided projection, while red arrows show the actual adjacent-level flip updates from W_q to W_q' . CLC reduces the first-moment shift directly in weight space under a flip budget.

3.2 The in-lattice correction

We begin with the local geometry of scalar quantization. For a weight $W_{i,j}$ with step size s_j , quantization maps the normalized value $W_{i,j}/s_j$ to one of the two adjacent integers bracketing it; nearest-level projection chooses the closer one. A complementary post-projection correction can only revise this local decision by moving the projected value to the adjacent lattice level on the other side. With residual $e_{i,j} = W_{q,i,j} - W_{i,j}$, this one-level revision is $\Delta_{i,j} = -\text{sign}(e_{i,j})s_j$ for a revised entry (i,j) , and $\Delta_{i,j} = 0$ otherwise. Thus, the correction remains inside the original lattice and only decides which local rounding decisions to revise.

The following two structural assumptions are used in the subsequent analysis. Both match common practice in practical weight quantization. The common-support assumption formalizes that the coordinates selected for correction come from a shared admissible set of layer input dimensions, while the per-channel step assumption corresponds to the standard setting in quantization methods where each output channel is quantized with its own uniform lattice step.

Assumption 1 (Common support). Letting $S_j := \{i : \Delta_{i,j} \neq 0\}$ be the per-channel support, all supports lie in a common index set: $S_j \subseteq \mathcal{I} \subseteq \{1, \dots, d\}$ for every j .

Assumption 2 (Uniform per-channel step). Quan-

tization uses a single step s_j per output channel, so the lattice $s_j\mathbb{Z}$ governing channel j is shared by all its weights.

Per-flip first-moment change. We now ask how a single one-level revision affects the mean-shift term. For channel j , recall $\hat{b}_j := \hat{\mu}^\top e_j$ and $\hat{B}_j := \hat{b}_j^2$. For a candidate revision at coordinate (i,j) , define $\Delta\hat{B}_{i,j}$ as the change in $\hat{B}_j$ caused by applying that revision alone. Since the revision changes $\hat{b}_j$ by $\hat{\mu}_i\Delta_{i,j}$ (see derivation in Appendix A), under $\Delta_{i,j} = -\text{sign}(e_{i,j})s_j$ we obtain:

$$\Delta\hat{B}_{i,j} = -2\hat{b}_j \text{sign}(e_{i,j})\hat{\mu}_i s_j + \hat{\mu}_i^2 s_j^2, \quad (3)$$

as derived in Appendix B. The linear term reduces the squared first-moment shift precisely when

$$\text{sign}(\hat{\mu}_i) = \text{sign}(e_{i,j}\hat{b}_j). \quad (4)$$

Under this sign alignment, $\Delta\hat{B}_{i,j} = -2|\hat{b}_j||\hat{\mu}_i|s_j + \hat{\mu}_i^2 s_j^2$, so an individual revision decreases the squared shift only if $2|\hat{b}_j| > |\hat{\mu}_i|s_j$. Hence Eq. (4) is a sign-alignment filter rather than a per-flip descent guarantee: large $|\hat{\mu}_i|s_j$ can still overshoot.

This observation suggests two requirements for an effective quantization correction. The candidate orientation and sign filter identify local in-lattice moves that can decrease the per-channel squared shift $\hat{B}_j = \hat{b}_j^2$, and hence contribute to reducing the layer mean-shift term $\hat{B}_\ell = \sum_j \hat{B}_j$ in Eq. (2).

Since an individual move can still overshoot, however, descent must be enforced after aggregation over the selected moves. We therefore seek a correction that yields a reduction in mean-shift error, $\hat{G}_\ell := \hat{\mathcal{B}}_\ell(W_q) - \hat{\mathcal{B}}_\ell(W'_q) \geq 0$, with the no-flip option kept feasible by the selection rule. We next ask how much variance cost these in-lattice moves can introduce.

3.3 A worst-case bound on the MSE change

The previous subsection shows how to obtain a non-negative mean-shift gain. For the population objective, write $G_\ell := \mathcal{B}_\ell(W_q) - \mathcal{B}_\ell(W'_q)$. However, reducing $\mathcal{B}_\ell$ alone does not guarantee a smaller total layer error, because the same lattice moves also change $\mathcal{V}_\ell$. Expanding (2) gives (See Appendix C):

$$\Delta\mathcal{R}_\ell = \underbrace{-G_\ell}_{\leq 0} + \underbrace{2\langle \Delta, \Sigma e \rangle_F + \text{tr}(\Delta^\top \Sigma \Delta)}_{\Delta\mathcal{V}_\ell}, \quad (5)$$

for any correction Δ and activation statistics (μ, Σ) . Thus the remaining question is how large the variance-side cost $\Delta\mathcal{V}_\ell$ can be.

The one-level rule in Section 3.2 gives a useful sign structure. For every selected move, $\Delta_{i,j} = -\text{sign}(e_{i,j})s_j$, so the diagonal part of the cross-term, $\sum_{i,j} \Delta_{i,j} \Sigma_{ii} e_{i,j}$, is non-positive because $\Delta_{i,j} \Sigma_{ii} e_{i,j} = -s_j \Sigma_{ii} |e_{i,j}| \leq 0$. Thus, this diagonal term can only reduce the variance-side change. Any possible increase in $\Delta\mathcal{V}_\ell$ must come from the off-diagonal covariance interactions, $\sum_j \sum_{i \neq k} \Delta_{i,j} \Sigma_{ik} e_{k,j}$, and the quadratic trace term, $\text{tr}(\Delta^\top \Sigma \Delta)$. Assumption 1 restricts these terms to the common support $\mathcal{I}$, and Assumption 2 gives a single lattice step per output channel. Together, they yield the following bound.

Theorem 1 (Worst-case MSE change). *Let Δ follow the one-level correction rule in Section 3.2 and satisfy Assumptions 1-2. Let $K_\ell := \sum_j |S_j|$ be the number of flips. Then*

$$\Delta\mathcal{R}_\ell \leq -G_\ell + K_\ell \Phi_\mathcal{I}, \quad (6)$$

where $\Phi_\mathcal{I} := -2\gamma_S + s_{\max}^2(\rho(\Sigma) + \|\Sigma_{\mathcal{I}\mathcal{I}}\|_2)$, $\gamma_S := \min_{(i,j): i \in S_j} s_j \Sigma_{ii} |e_{i,j}| \geq 0$, $s_{\max} := \max_j s_j$, and $\rho(\Sigma)$ is a worst-case off-diagonal row-sum of Σ . The proof is given in Appendix C.

Theorem 1 separates the effect of CLC into a mean-shift gain G_ℓ and a worst-case variance cost that grows at most linearly with the flip count K_ℓ . Hence, a safe correction should keep the support $\mathcal{I}$ covariance-friendly and limit K_ℓ so that the variance cost cannot dominate the gain.

Corollary 1 (The budget certifies non-increasing MSE). *Under the hypotheses of Theorem 1, if $\Phi_\mathcal{I} \leq 0$, then $\Delta\mathcal{R}_\ell \leq -G_\ell - K_\ell |\Phi_\mathcal{I}| \leq 0$ for any $K_\ell \geq 0$. If $\Phi_\mathcal{I} > 0$, then $\Delta\mathcal{R}_\ell \leq 0$ whenever $K_\ell \leq G_\ell / \Phi_\mathcal{I}$. The proof is given in Appendix D.*

Corollary 1 turns the bound into a budget rule. If $\Phi_\mathcal{I} \leq 0$, the variance side is harmless in the worst case. If $\Phi_\mathcal{I} > 0$, each flip may carry a positive variance cost, so the realized flip count must be small enough for the mean-shift gain to cover this cost. Thus, a safe selection rule should keep the no-flip option feasible, enforce non-negative aggregate mean-shift gain, and respect the certified budget. This certificate is intentionally worst-case; Appendix E.1 empirically verifies the conditions and measures how conservative the bound is on real layers.

3.4 The CLC algorithm

CLC instantiates the requirements derived above. It must choose a stable activation-mean estimate, define a support $\mathcal{I}$ that avoids unstable flips and keeps the covariance penalty controlled, enforce non-negative aggregate mean-shift gain, and respect the budget condition of Corollary 1. The overall framework of CLC is illustrated in Figure 2.

Stabilized activation mean. CLC relies on the calibration mean to estimate the per-channel shift $\hat{b}_j$ and to choose the input coordinates eligible for correction. Because these decisions are sensitive to coordinate-wise extremes in the empirical mean $\bar{x} = m^{-1} X \mathbf{1}_m$, we use the James-Stein shrinkage estimator (James and Stein, 1961), which stabilizes a high-dimensional mean by shrinking it toward a common center:

$$\begin{aligned} \hat{\mu} &= \bar{x}_g \mathbf{1} + (1 - c)(\bar{x} - \bar{x}_g \mathbf{1}), \\ c &= \text{clip}\left(\frac{(d-2)\hat{\sigma}^2}{\|\bar{x} - \bar{x}_g \mathbf{1}\|_2^2}, 0, 1\right), \end{aligned} \quad (7)$$

where $\bar{x}_g := d^{-1} \sum_i \bar{x}_i$ and $\hat{\sigma}^2 := (dm)^{-1} \sum_{i,t} (X_{i,t} - \bar{x}_i)^2$ is a pooled activation-variance estimate. Clipping prevents over-shrinkage and recovers the sample mean when the shrinkage estimate degenerates. All subsequent selection steps use this stabilized mean.

Support $\mathcal{I}$: a knee-point mask on $|\hat{\mu}|$. The support is motivated by the overshoot regime in (3). A coordinate with large $|\hat{\mu}_i|$ can make the quadratic term $\hat{\mu}_i^2 s_j^2$ dominate the linear mean-shift decrease

$2|\hat{b}_j||\hat{\mu}_i|s_j$, so flipping such coordinates may overshoot the channel shift. We therefore remove outlying $|\hat{\mu}_i|$ coordinates before per-channel selection. This choice also supports the MSE bound in Theorem 1: in LLM activations, large-mean coordinates often coincide with high-variance directions (Lin et al., 2024), so removing them can reduce the covariance penalty associated with $\|\Sigma_{\mathcal{I}\mathcal{I}}\|_2$.

Concretely, let $a_i := |\hat{\mu}_i|$ and sort the values in descending order, $a_{(1)} \geq \dots \geq a_{(d)}$. We apply a Kneedle-style rule (Satopaa et al., 2011) to the upper half of this sorted curve. After normalization, we connect the largest value $a_{(1)}$ to the mid-rank value $a_{(\lfloor d/2 \rfloor)}$ and choose the point with maximum distance from this line as the knee. The knee value defines the threshold τ , and the admissible support is $\mathcal{I} := \{i : |\hat{\mu}_i| \leq \tau\}$.

Filtered budgeted selection. For each channel j , candidates are first restricted to $\mathcal{I}$ and then passed through the sign-alignment filter from Section 3.2, giving $S_j \subseteq \mathcal{I}_j \subseteq \mathcal{I}$. A single hyperparameter $p \in (0, 1]$ sets the per-channel cap $B_j := \lceil p|\mathcal{I}_j| \rceil$, so the realized flip count in Theorem 1 satisfies $K_\ell = \sum_j |S_j| \leq \sum_j B_j$.

Within each channel, candidates are ordered by decreasing residual magnitude $\delta_{i,j} := |e_{i,j}|$. This prioritizes weights closest to the rounding boundary, for which the post-flip residual $|e'_{i,j}| = s_j - |e_{i,j}|$ is smallest. Define the signed reduction in calibration mean shift from flipping coordinate (i, j) as $v_{i,j} := -\hat{\mu}_i \Delta_{i,j} = \hat{\mu}_i \text{sign}(e_{i,j})s_j$. Let $n_j := |\mathcal{I}_j|$, and write the ordered candidates as $(i_1, \dots, i_{n_j})$. CLC then selects the best prefix under the budget:

$$k^* = \arg \min_{0 \leq k \leq B_j} \left| \hat{b}_j^{(k)} \right|, \quad \hat{b}_j^{(k)} := \hat{b}_j - \sum_{t=1}^k v_{i_t, j},$$

$$S_j := \{i_1, \dots, i_{k^*}\}.$$

(8)

Here B_j is only an upper bound: the selected prefix may stop earlier when additional flips would overshoot the channel shift. The minimizer is found by a single cumulative scan over the prefixes.

Calibration mean-shift descent. The descent guarantee is aggregate, not per-flip. Since the prefix search in (8) includes $k = 0$, the uncorrected channel is always feasible. Therefore $|\hat{b}_j| = |\hat{b}_j^{(k^*)}| \leq |\hat{b}_j^{(0)}| = |\hat{b}_j|$, and summing over channels gives

$$\widehat{G}_\ell := \widehat{B}_\ell(W_q) - \widehat{B}_\ell(W'_q) \geq 0.$$

This establishes the calibration analogue of the mean-shift gain used in Section 3.3; the formal proof is given in Appendix E. Algorithm 1 in Appendix F summarizes the procedure.

This analysis treats CLC as a correction step on a quantized lattice: given the stored codes and activation statistics in the same coordinate system, CLC applies adjacent-level updates to reduce the first-moment output shift. Appendix G discusses this interface for representative PTQ pipelines. For transformation-based methods such as AWQ, CLC can operate after the transformation is absorbed, using the transformed weights and activations. For Hessian-aware weight-update methods such as GPTQ, the mean-shift target remains tied to the original layer, while the admissible flip direction should follow the method-specific pre-rounding state. Although CLC is not designed for activation quantization, it can still be applied to weight-activation methods such as FlatQuant (Sun et al., 2025) by working in their transformed coordinate space. In contrast, methods with a broader quantization scope, such as SpinQuant (Liu et al., 2025a) with KV-cache quantization, require accounting for error interactions beyond the weight-induced layer output error considered here.

4 Experiments

Experimental setup. We evaluate CLC on four LLMs: Mistral-7B-v0.3 (Jiang et al., 2023), LLaMA-3.1-8B (Grattafiori et al., 2024), Qwen2.5-7B (Qwen et al., 2025), and LLaMA-3-8B (Grattafiori et al., 2024), at both 4-bit and 3-bit precision. To evaluate CLC across representative quantization paradigms, we use three weight-only baselines covering plain nearest-level projection (RTN), Hessian-aware error compensation (GPTQ (Frantar et al., 2023)), and activation-aware scaling (AWQ (Lin et al., 2024)). We also combine CLC with FlatQuant (Sun et al., 2025), a recent state-of-the-art transformation-based method for weight-activation quantization, to further validate the effectiveness of CLC. We also compare with classical bias correction (BC) (Nagel et al., 2019).

The calibration used is C4 with 128 sequences of length 2048 (Frantar et al., 2023). We report WikiText-2/C4 perplexity, zero-shot accuracy on ARC-Challenge, ARC-Easy, BoolQ, PIQA, and RTE (Sutawika et al., 2024), and their unweighted average; lower perplexity and higher accuracy are better. Due to space limits, the main paper

Table 1: 3-bit quantization results on Mistral-7B and LLaMA-3.1-8B. PPL on WikiText-2 and C4 ($\downarrow$); zero-shot accuracy on five tasks and their average ($\uparrow$). Bold marks the best non-FP16 row within each (model, base) block. Green shading marks the best non-FP16 result within each base-quantizer block, while red shading marks metrics degraded relative to the corresponding original quantizer.

Model	Method	PPL ($\downarrow$)		Downstream Accuracy ($\uparrow$)						
		Wiki	C4	arc-c	arc-e	boolq	piqa	rte	Avg	
Mistral-7B	FP16	4.845 $\pm$ 0.001	7.535 $\pm$ 0.002	0.509 $\pm$ 0.002	0.802 $\pm$ 0.001	0.837 $\pm$ 0.003	0.800 $\pm$ 0.003	0.686 $\pm$ 0.004	0.727	
	RTN	5.680 $\pm$ 0.005	8.883 $\pm$ 0.006	0.417 $\pm$ 0.003	0.663 $\pm$ 0.002	0.774 $\pm$ 0.002	0.773 $\pm$ 0.003	0.621 $\pm$ 0.001	0.650	
	RTN + BC	5.682 $\pm$ 0.006	8.887 $\pm$ 0.008	0.420 $\pm$ 0.003	0.664 $\pm$ 0.004	0.772 $\pm$ 0.002	0.788 $\pm$ 0.002	0.634 $\pm$ 0.004	0.656	
	RTN + CLC	5.639 $\pm$ 0.007	8.740 $\pm$ 0.005	0.422 $\pm$ 0.001	0.665 $\pm$ 0.003	0.788 $\pm$ 0.002	0.798 $\pm$ 0.002	0.676 $\pm$ 0.003	0.670	
	AWQ	5.572 $\pm$ 0.001	8.648 $\pm$ 0.002	0.439 $\pm$ 0.003	0.763 $\pm$ 0.003	0.780 $\pm$ 0.002	0.787 $\pm$ 0.001	0.603 $\pm$ 0.001	0.675	
	AWQ + BC	5.566 $\pm$ 0.003	8.633 $\pm$ 0.004	0.437 $\pm$ 0.002	0.760 $\pm$ 0.002	0.785 $\pm$ 0.003	0.787 $\pm$ 0.003	0.606 $\pm$ 0.001	0.675	
	AWQ + CLC	5.524 $\pm$ 0.005	8.552 $\pm$ 0.006	0.440 $\pm$ 0.003	0.766 $\pm$ 0.004	0.794 $\pm$ 0.002	0.787 $\pm$ 0.003	0.617 $\pm$ 0.001	0.681	
	GPTQ	5.424 $\pm$ 0.007	8.308 $\pm$ 0.008	0.449 $\pm$ 0.004	0.764 $\pm$ 0.002	0.785 $\pm$ 0.002	0.781 $\pm$ 0.002	0.667 $\pm$ 0.002	0.689	
	GPTQ + BC	5.418 $\pm$ 0.007	8.302 $\pm$ 0.009	0.449 $\pm$ 0.002	0.764 $\pm$ 0.003	0.781 $\pm$ 0.003	0.783 $\pm$ 0.002	0.664 $\pm$ 0.002	0.688	
	GPTQ + CLC	5.382 $\pm$ 0.004	8.277 $\pm$ 0.007	0.453 $\pm$ 0.003	0.774 $\pm$ 0.004	0.808 $\pm$ 0.001	0.790 $\pm$ 0.002	0.671 $\pm$ 0.004	0.699	
	FlatQuant	5.622 $\pm$ 0.003	8.566 $\pm$ 0.004	0.421 $\pm$ 0.003	0.759 $\pm$ 0.001	0.790 $\pm$ 0.003	0.789 $\pm$ 0.003	0.581 $\pm$ 0.003	0.668	
	FlatQuant + BC	5.901 $\pm$ 0.005	8.991 $\pm$ 0.006	0.417 $\pm$ 0.002	0.750 $\pm$ 0.002	0.791 $\pm$ 0.004	0.785 $\pm$ 0.003	0.598 $\pm$ 0.002	0.668	
	FlatQuant + CLC	5.587 $\pm$ 0.004	8.524 $\pm$ 0.004	0.433 $\pm$ 0.003	0.759 $\pm$ 0.002	0.788 $\pm$ 0.004	0.791 $\pm$ 0.002	0.607 $\pm$ 0.002	0.676	
LLaMA-3.1-8B	FP16	5.522 $\pm$ 0.005	8.409 $\pm$ 0.006	0.522 $\pm$ 0.003	0.822 $\pm$ 0.004	0.830 $\pm$ 0.001	0.791 $\pm$ 0.002	0.711 $\pm$ 0.003	0.735	
	RTN	10.947 $\pm$ 0.008	18.044 $\pm$ 0.013	0.386 $\pm$ 0.003	0.649 $\pm$ 0.002	0.717 $\pm$ 0.001	0.743 $\pm$ 0.002	0.588 $\pm$ 0.002	0.617	
	RTN + BC	10.132 $\pm$ 0.013	16.994 $\pm$ 0.014	0.359 $\pm$ 0.002	0.675 $\pm$ 0.003	0.647 $\pm$ 0.003	0.734 $\pm$ 0.002	0.567 $\pm$ 0.003	0.596	
	RTN + CLC	8.731 $\pm$ 0.009	14.168 $\pm$ 0.016	0.410 $\pm$ 0.003	0.729 $\pm$ 0.003	0.694 $\pm$ 0.004	0.756 $\pm$ 0.003	0.610 $\pm$ 0.003	0.640	
	AWQ	8.094 $\pm$ 0.007	13.157 $\pm$ 0.006	0.413 $\pm$ 0.002	0.745 $\pm$ 0.002	0.715 $\pm$ 0.002	0.756 $\pm$ 0.001	0.624 $\pm$ 0.004	0.651	
	AWQ + BC	8.053 $\pm$ 0.005	13.143 $\pm$ 0.009	0.410 $\pm$ 0.002	0.736 $\pm$ 0.003	0.719 $\pm$ 0.002	0.759 $\pm$ 0.003	0.632 $\pm$ 0.003	0.651	
	AWQ + CLC	7.893 $\pm$ 0.007	12.851 $\pm$ 0.007	0.417 $\pm$ 0.002	0.749 $\pm$ 0.001	0.724 $\pm$ 0.002	0.764 $\pm$ 0.002	0.638 $\pm$ 0.003	0.658	
	GPTQ	7.220 $\pm$ 0.003	11.566 $\pm$ 0.004	0.413 $\pm$ 0.002	0.739 $\pm$ 0.004	0.767 $\pm$ 0.004	0.760 $\pm$ 0.003	0.703 $\pm$ 0.003	0.676	
	GPTQ + BC	7.221 $\pm$ 0.005	11.567 $\pm$ 0.002	0.412 $\pm$ 0.001	0.737 $\pm$ 0.003	0.766 $\pm$ 0.003	0.761 $\pm$ 0.003	0.703 $\pm$ 0.003	0.676	
	GPTQ + CLC	7.215 $\pm$ 0.003	11.560 $\pm$ 0.002	0.415 $\pm$ 0.001	0.741 $\pm$ 0.002	0.767 $\pm$ 0.004	0.762 $\pm$ 0.002	0.705 $\pm$ 0.003	0.678	
	FlatQuant	7.443 $\pm$ 0.003	11.924 $\pm$ 0.003	0.421 $\pm$ 0.003	0.756 $\pm$ 0.001	0.735 $\pm$ 0.003	0.768 $\pm$ 0.002	0.628 $\pm$ 0.002	0.662	
	FlatQuant + BC	7.448 $\pm$ 0.001	12.095 $\pm$ 0.002	0.420 $\pm$ 0.002	0.753 $\pm$ 0.002	0.746 $\pm$ 0.001	0.766 $\pm$ 0.003	0.638 $\pm$ 0.003	0.665	
	FlatQuant + CLC	7.381 $\pm$ 0.003	11.881 $\pm$ 0.004	0.415 $\pm$ 0.002	0.760 $\pm$ 0.002	0.750 $\pm$ 0.001	0.768 $\pm$ 0.002	0.661 $\pm$ 0.003	0.671	

reports Mistral-7B-v0.3 and LLaMA-3.1-8B; results on Qwen2.5-7B and LLaMA-3-8B, additional AdaRound (Nagel et al., 2020) experiments, flip-budget sensitivity, and ablations are deferred to Appendix H. AdaRound further tests whether CLC extends to learned-rounding integer-lattice pipelines beyond nearest-style projection.

4.1 Main Results

Tables 1 and 2 report the 3-bit and 4-bit results, respectively, using the same models, base quantizers, and downstream tasks. Bold denotes the best non-FP16 row within each (model, base) block.

CLC consistently improves perplexity. Tables 1 and 2 show that CLC reduces perplexity across all main model-quantizer blocks. At 4-bit, the gains are modest but consistent, as the base quantizers are already close to FP16; for example, CLC improves Mistral-7B AWQ from 4.975 to 4.965 on WikiText-2 and from 7.727 to 7.711 on C4. At 3-bit, where the integer lattice is coarser, the improvements become much larger. On LLaMA-3.1-8B with RTN, CLC reduces WikiText-2 PPL from 10.947 to 8.731 and C4 PPL from 18.044 to 14.168. Similar gains

Mistral-7B Quantization Runtime: Original vs CLC Applied

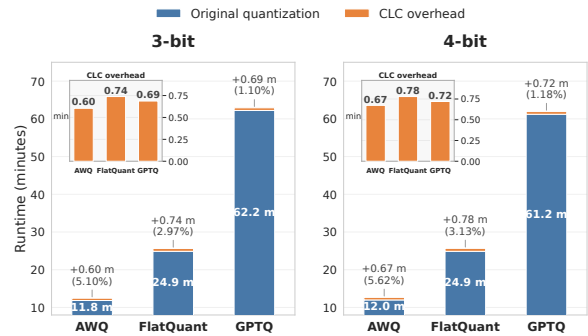


Figure 3: Runtime comparison on Mistral-7B. CLC adds a small post-processing cost on top of the base quantizer, while the total runtime remains dominated by the original quantization stage.

on other baselines indicate that CLC is not limited to plain RTN, but corrects a residual left by the final discrete projection of different PTQ pipelines.

CLC is more reliable than bias correction. Compared with classical bias correction (BC), CLC provides more reliable improvements across the main settings. BC does not always improve over the original quantized model; for example,

Table 2: 4-bit quantization results on Mistral-7B and LLaMA-3.1-8B. PPL on WikiText-2 and C4 ($\downarrow$); zero-shot accuracy on five tasks and their average ($\uparrow$). Bold marks the best non-FP16 row within each (model, base) block. Green shading marks the best non-FP16 result within each base-quantizer block, while red shading marks metrics degraded relative to the corresponding original quantizer.

Model	Method	PPL ($\downarrow$)		Downstream Accuracy ($\uparrow$)					
		Wiki	C4	arc-c	arc-e	boolq	piqa	rte	Avg
Mistral-7B	FP16	4.845 $\pm$ 0.001	7.535 $\pm$ 0.002	0.509 $\pm$ 0.002	0.802 $\pm$ 0.001	0.837 $\pm$ 0.003	0.800 $\pm$ 0.003	0.686 $\pm$ 0.004	0.727
	RTN	4.992 $\pm$ 0.003	7.805 $\pm$ 0.004	0.476 $\pm$ 0.001	0.787 $\pm$ 0.004	0.825 $\pm$ 0.002	0.799 $\pm$ 0.004	0.664 $\pm$ 0.002	0.710
	RTN + BC	4.995 $\pm$ 0.005	7.738 $\pm$ 0.006	0.474 $\pm$ 0.003	0.789 $\pm$ 0.002	0.828 $\pm$ 0.002	0.799 $\pm$ 0.003	0.661 $\pm$ 0.003	0.710
	RTN + CLC	4.970 $\pm$ 0.007	7.717 $\pm$ 0.005	0.480 $\pm$ 0.003	0.793 $\pm$ 0.002	0.820 $\pm$ 0.003	0.801 $\pm$ 0.001	0.675 $\pm$ 0.003	0.714
	AWQ	4.975 $\pm$ 0.003	7.727 $\pm$ 0.003	0.493 $\pm$ 0.003	0.794 $\pm$ 0.002	0.817 $\pm$ 0.002	0.797 $\pm$ 0.003	0.671 $\pm$ 0.004	0.714
	AWQ + BC	4.975 $\pm$ 0.001	7.726 $\pm$ 0.002	0.493 $\pm$ 0.002	0.793 $\pm$ 0.004	0.815 $\pm$ 0.004	0.798 $\pm$ 0.004	0.678 $\pm$ 0.003	0.715
	AWQ + CLC	4.965 $\pm$ 0.003	7.711 $\pm$ 0.004	0.495 $\pm$ 0.004	0.795 $\pm$ 0.003	0.818 $\pm$ 0.002	0.802 $\pm$ 0.002	0.690 $\pm$ 0.004	0.720
	GPTQ	4.978 $\pm$ 0.005	7.744 $\pm$ 0.005	0.480 $\pm$ 0.002	0.795 $\pm$ 0.002	0.820 $\pm$ 0.003	0.798 $\pm$ 0.002	0.675 $\pm$ 0.002	0.714
	GPTQ + BC	4.979 $\pm$ 0.007	7.745 $\pm$ 0.004	0.480 $\pm$ 0.002	0.795 $\pm$ 0.004	0.819 $\pm$ 0.004	0.798 $\pm$ 0.004	0.675 $\pm$ 0.003	0.713
	GPTQ + CLC	4.967 $\pm$ 0.004	7.736 $\pm$ 0.005	0.483 $\pm$ 0.003	0.796 $\pm$ 0.001	0.821 $\pm$ 0.002	0.800 $\pm$ 0.002	0.678 $\pm$ 0.003	0.716
	FlatQuant	5.072 $\pm$ 0.005	7.854 $\pm$ 0.005	0.483 $\pm$ 0.003	0.794 $\pm$ 0.002	0.829 $\pm$ 0.002	0.794 $\pm$ 0.004	0.660 $\pm$ 0.002	0.712
	FlatQuant + BC	5.073 $\pm$ 0.003	7.851 $\pm$ 0.004	0.468 $\pm$ 0.002	0.787 $\pm$ 0.001	0.825 $\pm$ 0.003	0.795 $\pm$ 0.002	0.653 $\pm$ 0.003	0.706
	FlatQuant + CLC	5.066 $\pm$ 0.005	7.847 $\pm$ 0.003	0.491 $\pm$ 0.004	0.793 $\pm$ 0.002	0.830 $\pm$ 0.001	0.797 $\pm$ 0.003	0.675 $\pm$ 0.002	0.717
LLaMA-3.1-8B	FP16	5.522 $\pm$ 0.005	8.409 $\pm$ 0.006	0.522 $\pm$ 0.003	0.822 $\pm$ 0.004	0.830 $\pm$ 0.001	0.791 $\pm$ 0.002	0.711 $\pm$ 0.003	0.735
	RTN	6.025 $\pm$ 0.007	9.440 $\pm$ 0.008	0.490 $\pm$ 0.004	0.806 $\pm$ 0.004	0.799 $\pm$ 0.003	0.783 $\pm$ 0.002	0.635 $\pm$ 0.001	0.703
	RTN + BC	6.082 $\pm$ 0.006	9.631 $\pm$ 0.007	0.487 $\pm$ 0.003	0.801 $\pm$ 0.001	0.808 $\pm$ 0.002	0.785 $\pm$ 0.002	0.632 $\pm$ 0.001	0.702
	RTN + CLC	5.973 $\pm$ 0.005	9.369 $\pm$ 0.005	0.505 $\pm$ 0.002	0.807 $\pm$ 0.004	0.793 $\pm$ 0.002	0.787 $\pm$ 0.003	0.653 $\pm$ 0.003	0.709
	AWQ	5.968 $\pm$ 0.005	9.321 $\pm$ 0.006	0.511 $\pm$ 0.003	0.806 $\pm$ 0.003	0.794 $\pm$ 0.001	0.798 $\pm$ 0.004	0.635 $\pm$ 0.003	0.709
	AWQ + BC	5.968 $\pm$ 0.008	9.316 $\pm$ 0.007	0.507 $\pm$ 0.002	0.803 $\pm$ 0.004	0.790 $\pm$ 0.001	0.796 $\pm$ 0.002	0.632 $\pm$ 0.002	0.706
	AWQ + CLC	5.949 $\pm$ 0.004	9.294 $\pm$ 0.005	0.512 $\pm$ 0.002	0.808 $\pm$ 0.002	0.796 $\pm$ 0.001	0.796 $\pm$ 0.003	0.671 $\pm$ 0.004	0.717
	GPTQ	5.995 $\pm$ 0.002	9.304 $\pm$ 0.001	0.505 $\pm$ 0.003	0.804 $\pm$ 0.001	0.795 $\pm$ 0.003	0.793 $\pm$ 0.004	0.651 $\pm$ 0.001	0.710
	GPTQ + BC	5.992 $\pm$ 0.003	9.304 $\pm$ 0.004	0.499 $\pm$ 0.002	0.803 $\pm$ 0.003	0.792 $\pm$ 0.004	0.793 $\pm$ 0.001	0.659 $\pm$ 0.003	0.709
	GPTQ + CLC	5.989 $\pm$ 0.001	9.300 $\pm$ 0.002	0.507 $\pm$ 0.003	0.806 $\pm$ 0.003	0.790 $\pm$ 0.003	0.797 $\pm$ 0.003	0.663 $\pm$ 0.003	0.713
	FlatQuant	6.158 $\pm$ 0.003	9.816 $\pm$ 0.004	0.488 $\pm$ 0.001	0.797 $\pm$ 0.003	0.801 $\pm$ 0.002	0.781 $\pm$ 0.004	0.657 $\pm$ 0.004	0.705
	FlatQuant + BC	6.164 $\pm$ 0.005	9.841 $\pm$ 0.003	0.486 $\pm$ 0.003	0.793 $\pm$ 0.002	0.808 $\pm$ 0.002	0.786 $\pm$ 0.001	0.662 $\pm$ 0.002	0.707
	FlatQuant + CLC	6.147 $\pm$ 0.004	9.813 $\pm$ 0.003	0.487 $\pm$ 0.002	0.805 $\pm$ 0.002	0.809 $\pm$ 0.003	0.787 $\pm$ 0.002	0.668 $\pm$ 0.004	0.711

at 4-bit on LLaMA-3.1-8B with RTN, BC worsens WikiText-2/C4 perplexity from 6.025/9.440 to 6.082/9.631, while CLC improves them to 5.973/9.369. This suggests that directly correcting the integer-lattice residual is a more stable post-quantization correction than compensating the shift only through an output-side bias term.

Downstream accuracy improves on average. CLC also improves the average zero-shot score in every main model-quantizer block. The gains are most visible in harder 3-bit settings, such as Mistral-7B with GPTQ, where average accuracy improves from 0.689 to 0.699, and LLaMA-3.1-8B with RTN, where it improves from 0.617 to 0.640. Individual tasks may fluctuate, but the aggregate trend is consistently positive. Overall, CLC acts as a lightweight post-quantization correction that can be attached to RTN, AWQ, GPTQ, and FlatQuant without changing the inference architecture.

CLC is complementary to different quantizers. CLC improves plain RTN and transformation-based quantizers such as AWQ and FlatQuant, suggesting that it corrects residual error left by the

final discrete projection rather than replacing the base quantizer’s objective. The gains are largest when this residual is more severe, especially at 3-bit and in RTN-like settings, but remain visible for stronger 4-bit quantizers. On GPTQ, the improvements are consistent but more modest, particularly on LLaMA-3.1-8B, because GPTQ already performs Hessian-aware error compensation. As discussed in Section 3.4 and Appendix G, CLC fits RTN and transformation-based pipelines more directly. GPTQ requires a more careful interface because its Hessian-aware updates define a method-specific pre-round state for admissible flips, while the mean-shift target is still measured against the final error of the original layer.

4.2 Computational Cost

CLC adds only a lightweight post-processing pass after the base quantizer has finished. Figure 3 reports the runtime on Mistral-7B for AWQ, FlatQuant, and GPTQ at both 3-bit and 4-bit. Across all settings, the additional CLC cost is below one minute, ranging from 0.60 to 0.78 minutes. This corresponds to only 1.10-5.62% of the orig-

inal quantization time. The relative overhead is largest for the fastest pipeline, AWQ, and smallest for the slowest pipeline, GPTQ, showing that the end-to-end runtime remains dominated by the base quantization procedure. Thus, CLC improves the final quantized model without retraining, architectural changes, or meaningful deployment overhead.

5 Conclusion

We introduced Complementary Lattice Correction (CLC), a post-quantization correction that reduces structured first-moment output error by making budgeted adjacent-level updates directly on the integer weight lattice. Unlike classical bias correction, CLC preserves the bias-less architecture and inference graph while providing an effective layer-wise descent mechanism for the quantization-induced mean shift. Experiments across bit-widths, models, and PTQ backbones show that CLC consistently improves perplexity and usually improves downstream accuracy with negligible post-processing overhead. We frame CLC as a lightweight post-hoc enhancement rather than a replacement quantizer: gains over strong 4-bit baselines are typically small given the limited remaining headroom, while larger gains appear in the more challenging 3-bit regime.

Limitations

Although CLC is effective across the evaluated settings, several aspects deserve further study. First, like other post-training methods, CLC uses activation statistics estimated from a limited calibration set. We reduce this sensitivity with James-Stein shrinkage and evaluate cross-dataset robustness in Section E.1, and we further show in Appendix I.2 that CLC’s gains are stable across calibration-set sizes; nevertheless, broader prompt distributions and context lengths remain useful settings for future evaluation. Second, modern PTQ pipelines expose increasingly diverse transformed spaces, rounding states, and error-compensation mechanisms. CLC can be attached as a lightweight post-quantization lattice refinement, but different quantizer families may benefit from tailored choices of coordinates, statistics, and admissible rounding moves. Developing a more general rounding-selection framework for diverse PTQ pipelines is an important direction for future work. Finally, our analysis focuses on the layer-wise effect of CLC, which matches how the correction is applied in practice. We measure the inter-

layer accumulation of the mean shift empirically in Appendix I.3 and find that CLC does not amplify cross-layer interactions, but a more detailed transformer-level analysis of how local corrections interact across layers—together with a theoretically guaranteed propagated-input objective—would further strengthen the theoretical understanding of CLC.

Ethical Considerations

We comply with the ACL Code of Ethics and all relevant license terms. Our research in quantization is designed to enhance the field. When applied responsibly, it carries no significant societal risks.

Acknowledgements

This research was funded by Vingroup Innovation Foundation (VINIF) under project code VINIF.2026.DA117.

References

- Yamato Arai and Yuma Ichikawa. 2025. [Quantization error propagation: Revisiting layer-wise post-training quantization](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefer, and James Hensman. 2024. [Quarot: Outlier-free 4-bit inference in rotated LLMs](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. [GPT3.int8\(\): 8-bit matrix multiplication for transformers at scale](#). In *Advances in Neural Information Processing Systems*.
- Tim Dettmers, Ruslan A. Svirshchevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefer, and Dan Alistarh. 2024. [SpQR: A sparse-quantized representation for near-lossless LLM weight compression](#). In *The Twelfth International Conference on Learning Representations*.
- Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. 2024. [Extreme compression of large language models via additive quantization](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 12284–12303. PMLR / OpenReview.net.
- Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. 2020. [Learned step size quantization](#). In

- International Conference on Learning Representations*.
- Elias Frantar and Dan Alistarh. 2022. [Optimal brain compression: A framework for accurate post-training quantization and pruning](#). In *Advances in Neural Information Processing Systems*.
- Elias Frantar, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. 2023. [Gptq: Accurate post-training quantization for generative pre-trained transformers](#). *Preprint*, arXiv:2210.17323.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Jung Hwan Heo, Jeonghoon Kim, Beomseok Kwon, Byeongwook Kim, Se Jung Kwon, and Dongsoo Lee. 2024. [Rethinking channel dimensions to isolate outliers for low-bit weight quantization of large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. [Quantization and training of neural networks for efficient integer-arithmetic-only inference](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- William James and Charles Stein. 1961. [Estimation with quadratic loss](#). In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 4.1, pages 361–379.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Sangil Jung, Changyong Son, Seohyung Lee, JinWoo Son, Jae-Joon Han, Youngjun Kwak, Sung Ju Hwang, and Changkyu Choi. 2019. [Learning to quantize deep networks by optimizing quantization intervals with task loss](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4350–4359.
- Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. 2024. [Squeezellm: Dense-and-sparse quantization](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 23901–23923. PMLR / OpenReview.net.
- Changhun Lee, Jungyu Jin, Taesu Kim, Hyungjun Kim, and Eunhyeok Park. 2024. [OWQ: outlier-aware weight quantization for efficient fine-tuning and inference of large language models](#). In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 13355–13364. AAAI Press.
- Janghwan Lee, Jiwoong Park, Jinseok Kim, Yongjik Kim, Jungju Oh, Jinwook Oh, and Jungwook Choi. 2025. [AMXFP4: Taming activation outliers with asymmetric microscaling floating-point for 4-bit LLM inference](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14993–15013, Vienna, Austria. Association for Computational Linguistics.
- Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. 2021. [BRECQ: pushing the limit of post-training quantization by block reconstruction](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. [Awq: Activation-aware weight quantization for on-device llm compression and acceleration](#). In *Proceedings of Machine Learning and Systems*, volume 6, pages 87–100.
- Shih-yang Liu, Zechun Liu, Xijie Huang, Pingcheng Dong, and Kwang-Ting Cheng. 2023. [LLM-FP4: 4-bit floating-point quantized transformers](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 592–605, Singapore. Association for Computational Linguistics.
- Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2024. [LLM-QAT: Data-free quantization aware training for large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 467–484, Bangkok, Thailand. Association for Computational Linguistics.
- Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. 2025a. [Spinquant: LLM quantization with learned rotations](#). In *The Thirteenth International Conference on Learning Representations*.
- Zechun Liu, Changsheng Zhao, Hanxian Huang, Sijia Chen, Jing Zhang, Jiawei Zhao, Scott Roy, Lisa Jin, Yunyang Xiong, Yangyang Shi, Lin Xiao, Yuandong Tian, Bilge Soran, Raghuraman Krishnamoorthi, Tijmen Blankevoort, and Vikas Chandra. 2025b. [Paretoq: Improving scaling laws in extremely low-bit](#)

- LLM quantization.** In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. 2020. **Up or down? adaptive rounding for post-training quantization.** In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, Proceedings of Machine Learning Research, pages 7197–7206. PMLR.
- Markus Nagel, Mart Van Baalen, Tijmen Blankevoort, and Max Welling. 2019. **Data-free quantization through weight equalization and bias correction.** In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1325–1334.
- Truong Nguyen, Phi Van Dat, Ngan Nguyen, Linh Ngo Van, Trung Le, and Thanh Hong Nguyen. 2026. **CTPD: cross tokenizer preference distillation.** In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 37783–37790. AAAI Press.
- Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. **Qwen2.5 technical report.** *Preprint*, arXiv:2412.15115.
- Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. 2011. **Finding a "kneedle" in a haystack: Detecting knee points in system behavior.** In *2011 31st International Conference on Distributed Computing Systems Workshops*, pages 166–171.
- Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. 2024. **Omniquant: Omnidirectionally calibrated quantization for large language models.** In *The Twelfth International Conference on Learning Representations*.
- Yuantian Shao, Yuanteng Chen, Peisong Wang, Jianlin Yu, Jing Lin, Yiwu Yao, Zhihui Wei, and Jian Cheng. 2025. **Dartquant: Efficient rotational distribution calibration for LLM quantization.** In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. 2025. **Flatquant: Flatness matters for LLM quantization.** In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.
- Lintang Sutawika, Hailey Schoelkopf, Leo Gao, Baber Abbasi, Stella Biderman, Jonathan Tow, ben fat-tori, Charles Lovering, farzanehnakhaee70, Jason Phang, Anish Thite, Fazz, Aflah, Niklas Muen-nighoff, Thomas Wang, sdtblck, nopperl, gakada, tttuntian, and 11 others. 2024. **Eleutherai/lm-evaluation-harness: v0.4.3.**
- Minh-Phuc Truong, Hai An Vu, Tu Vu, Nguyen Thi Ngoc Diep, Linh Ngo Van, Thien Huu Nguyen, and Trung Le. 2025. **EMO: Embedding model distillation via intra-model relation and optimal transport alignments.** In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7594–7606, Suzhou, China. Association for Computational Linguistics.
- Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. 2024. **Quip#: Even better LLM quantization with hadamard incoherence and lattice codebooks.** In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 48630–48656. PMLR / OpenReview.net.
- Duc Trung Vu, Pham Khanh Chi, Dat Phi Van, Linh Ngo Van, Dinh Viet Sang, and Trung Le. 2026. **DWA-KD: Dual-space weighting and time-warped alignment for cross-tokenizer knowledge distillation.** In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 3513–3527, Rabat, Morocco. Association for Computational Linguistics.
- Hoang Tran Vuong, Tue Le, Quyen Tran, Linh Ngo Van, and Trung Le. 2026. **MCW-KD: multi-cost wasserstein knowledge distillation for large language models.** In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 33332–33340. AAAI Press.
- Hongyu Wang, Shuming Ma, Lingxiao Ma, Lei Wang, Wenhui Wang, Li Dong, Shaohan Huang, Huaijie Wang, Jilong Xue, Ruiping Wang, Yi Wu, and Furu Wei. 2025. **Bitnet: 1-bit pre-training for large language models.** *Journal of Machine Learning Research*, 26(125):1–29.
- Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. 2023. **Outlier suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling.** In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1648–1665, Singapore. Association for Computational Linguistics.
- Guangxuan Xiao, Ji Lin, Mickaël Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. **Smoothquant: Accurate and efficient post-training quantization for large language models.** In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023*,

Honolulu, Hawaii, USA, Proceedings of Machine Learning Research, pages 38087–38099. PMLR.

Zhewei Yao, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael W. Mahoney, and Kurt Keutzer. 2021. [HAWQ-V3: dyadic neural network quantization](#). In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, pages 11875–11886. PMLR.

Zhewei Yao, Xiaoxia Wu, Cheng Li, Stephen Youn, and Yuxiong He. 2023. [Zeroquant-v2: Exploring post-training quantization in llms from comprehensive study to low rank compensation](#). *Preprint*, arXiv:2303.08302.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. [Zeroquant: Efficient and affordable post-training quantization for large-scale transformers](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 27168–27183. Curran Associates, Inc.

Xi Zhang, Xiaolin Wu, Jiamang Wang, and Weisi Lin. 2025. [Learning grouped lattice vector quantizers for low-bit llm compression](#). In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 110672–110698. Curran Associates, Inc.

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

A Derivation of the Layer-wise Error Decomposition

We derive the decomposition used in Eq. (2). Consider a single output channel j of a linear layer. Let

$$w_j := W_{:,j}, \quad w_{q,j} := (W_q)_{:,j}, \quad e_j := w_{q,j} - w_j.$$

For an activation vector $x \in \mathbb{R}^d$, the full-precision and quantized channel outputs are

$$y_j = x^\top w_j, \quad y_{q,j} = x^\top w_{q,j}.$$

Their difference is therefore

$$y_{q,j} - y_j = x^\top (w_{q,j} - w_j) = x^\top e_j.$$

The expected squared reconstruction error of this channel is

$$\mathcal{R}_j := \mathbb{E}[(y_{q,j} - y_j)^2] = \mathbb{E}[(x^\top e_j)^2]. \quad (9)$$

Since $(x^\top e_j)^2 = e_j^\top x x^\top e_j$, we have

$$\mathcal{R}_j = e_j^\top \mathbb{E}[x x^\top] e_j. \quad (10)$$

Now write the activation as

$$x = \mu + \varepsilon, \quad \mu := \mathbb{E}[x], \quad \mathbb{E}[\varepsilon] = 0,$$

and define the covariance matrix

$$\Sigma := \text{Cov}(x) = \mathbb{E}[(x - \mu)(x - \mu)^\top] = \mathbb{E}[\varepsilon \varepsilon^\top].$$

Then

$$\begin{aligned} \mathbb{E}[x x^\top] &= \mathbb{E}[(\mu + \varepsilon)(\mu + \varepsilon)^\top] \\ &= \mu \mu^\top + \mathbb{E}[\mu \varepsilon^\top] + \mathbb{E}[\varepsilon \mu^\top] + \mathbb{E}[\varepsilon \varepsilon^\top] \\ &= \mu \mu^\top + \Sigma. \end{aligned} \quad (11)$$

Substituting Eq. (11) into Eq. (10) gives

$$\begin{aligned} \mathcal{R}_j &= e_j^\top (\mu \mu^\top + \Sigma) e_j \\ &= (\mu^\top e_j)^2 + e_j^\top \Sigma e_j. \end{aligned} \quad (12)$$

Summing over all output channels yields

$$\begin{aligned} \mathcal{R}_\ell(W_q) &= \sum_{j=1}^C \mathbb{E}[(x^\top e_j)^2] \\ &= \underbrace{\sum_{j=1}^C (\mu^\top e_j)^2}_{\mathcal{B}_\ell} + \underbrace{\sum_{j=1}^C e_j^\top \Sigma e_j}_{\mathcal{V}_\ell}. \end{aligned} \quad (13)$$

Thus, the layer-wise reconstruction error decomposes exactly into a squared first-moment shift term $\mathcal{B}_\ell$ and a covariance-weighted term $\mathcal{V}_\ell$.

B Derivation of the Per-flip First-moment Change

We derive Eq. (3). For one output channel j , the calibration-evaluated first-moment shift is $\hat{b}_j := \hat{\mu}^\top e_j$, with squared contribution $\hat{B}_j := \hat{b}_j^2$. If only coordinate (i, j) is revised by $\Delta_{i,j}$, then under the residual convention $e_j = (W_q - W)_{:,j}$ the revised residual is $e_j + \Delta_{i,j} \mathbf{e}_i$, where $\mathbf{e}_i$ is the i -th standard basis vector. Thus

$$\hat{b}'_j = \hat{\mu}^\top (e_j + \Delta_{i,j} \mathbf{e}_i) = \hat{b}_j + \hat{\mu}_i \Delta_{i,j}.$$

Therefore,

$$\begin{aligned} \Delta \hat{B}_{i,j} &:= (\hat{b}'_j)^2 - \hat{b}_j^2 \\ &= (\hat{b}_j + \hat{\mu}_i \Delta_{i,j})^2 - \hat{b}_j^2 \\ &= 2\hat{b}_j \hat{\mu}_i \Delta_{i,j} + \hat{\mu}_i^2 \Delta_{i,j}^2. \end{aligned} \quad (14)$$

For a one-level revision, $\Delta_{i,j} = -\text{sign}(e_{i,j}) s_j$, so $\Delta_{i,j}^2 = s_j^2$. Substituting gives

$$\Delta \hat{B}_{i,j} = -2\hat{b}_j \text{sign}(e_{i,j}) \hat{\mu}_i s_j + \hat{\mu}_i^2 s_j^2.$$

The linear term is negative exactly when $\hat{b}_j \text{sign}(e_{i,j}) \hat{\mu}_i > 0$, equivalently $\text{sign}(\hat{\mu}_i) = \text{sign}(e_{i,j} \hat{b}_j)$. Under this condition,

$$\Delta \hat{B}_{i,j} = -2|\hat{b}_j| |\hat{\mu}_i| s_j + \hat{\mu}_i^2 s_j^2,$$

which is strictly negative if and only if $2|\hat{b}_j| > |\hat{\mu}_i| s_j$.

C Proof of Theorem 1

Throughout, $e := W_q - W$, $e_j := e_{:,j} \in \mathbb{R}^d$ is the residual of channel j , and $S_j := \{i : \Delta_{i,j} \neq 0\}$ its support. A CLC correction follows the one-level correction rule of Section 3.2 and satisfies Assumptions 1-2: each flipped entry is the anti-aligned unit flip $\Delta_{i,j} = -\text{sign}(e_{i,j}) s_j$ for $i \in S_j$ (zero otherwise), all supports lie inside a common $\mathcal{I}$, and a single per-channel step s_j governs each channel.

Proof plan. The argument has three stages. (1) We first establish the *exact* change in layer MSE under any lattice correction; this is an algebraic identity with no inequalities, splitting $\Delta \mathcal{R}_\ell$ into the bias gain $-G_\ell$ and a variance change $\Delta \mathcal{V}_\ell$. (2) We then decompose the cross-term inside $\Delta \mathcal{V}_\ell$ into a diagonal piece D_ℓ and an off-diagonal piece $\mathcal{E}_\ell$ (Lemma 1); the anti-aligned orientation forces

$D_\ell \leq 0$ deterministically, while $|\mathcal{E}_\ell|$ admits a row-sum bound from the RTN residual bound. (3) Finally we bound the trace term using the common support (Assumption 1) and the unit-flip norm, and assemble the three bounds into (6). Stage (1) is where the exact identity (5) of the body is proved; stages (2) and (3) are pure upper bounds, and their looseness is intentional (see the closing remark).

C.1 Stage 1: the exact identity

From the bias-variance decomposition (2), $\mathcal{R}_\ell(W_q) = \sum_j (\mu^\top e_j)^2 + e_j^\top \Sigma e_j$. Writing $e'_j := e_j + \Delta_j$ for the corrected residual of channel j (where $\Delta_j := \Delta_{:,j}$),

$$\begin{aligned} \mathcal{R}_\ell(W'_q) - \mathcal{R}_\ell(W_q) &= \sum_j \left[(\mu^\top e'_j)^2 - (\mu^\top e_j)^2 \right] \\ &\quad + \sum_j \left[e_j'^\top \Sigma e'_j - e_j^\top \Sigma e_j \right]. \end{aligned}$$

By the definition $G_\ell = \mathcal{B}_\ell(W_q) - \mathcal{B}_\ell(W'_q)$ the first sum is $-G_\ell$. The second sum expands as $2 \sum_j \langle \Delta_j, \Sigma e_j \rangle + \sum_j \Delta_j^\top \Sigma \Delta_j = 2 \langle \Delta, \Sigma e \rangle_F + \text{tr}(\Delta^\top \Sigma \Delta)$. Hence, for any Δ and any (μ, Σ) ,

$$\Delta \mathcal{R}_\ell = -G_\ell + 2 \langle \Delta, \Sigma e \rangle_F + \text{tr}(\Delta^\top \Sigma \Delta), \quad (15)$$

which is the exact decomposition (5) used in the body, with variance change $\Delta \mathcal{V}_\ell = 2 \langle \Delta, \Sigma e \rangle_F + \text{tr}(\Delta^\top \Sigma \Delta)$.

C.2 Stage 2: the cross-term

We use the constants $K_\ell := \sum_j |S_j|$ (total flip count), $s_{\max} := \max_j s_j$, $\rho(\Sigma) := \max_i \sum_{k \neq i} |\Sigma_{ik}|$ the worst-case off-diagonal row-sum of Σ , and $\gamma_S := \min_{(i,j): i \in S_j} s_j \sum_{ii} |e_{i,j}| \geq 0$ the smallest per-flip diagonal gain over the realized support $\{S_j\}$.

For each channel j , decompose $(\Sigma e_j)_i = \Sigma_{ii} e_{i,j} + \sum_{k \neq i} \Sigma_{ik} e_{k,j}$ and substitute the anti-aligned flip rule $\Delta_{i,j} = -\text{sign}(e_{i,j}) s_j$ from the one-level correction rule of Section 3.2 into $\langle \Delta_j, \Sigma e_j \rangle = \sum_{i \in S_j} \Delta_{i,j} (\Sigma e_j)_i$. Using $\text{sign}(e_{i,j}) e_{i,j} = |e_{i,j}|$, this yields the diagonal piece

$$D_j := -s_j \sum_{i \in S_j} \Sigma_{ii} |e_{i,j}|, \quad (16)$$

and the off-diagonal piece

$$\mathcal{E}_j := -s_j \sum_{i \in S_j} \text{sign}(e_{i,j}) \sum_{k \neq i} \Sigma_{ik} e_{k,j}, \quad (17)$$

so that $\langle \Delta_j, \Sigma e_j \rangle = D_j + \mathcal{E}_j$.

Lemma 1 (Cross-term sign and bound). *Let D_j and $\mathcal{E}_j$ be as defined in (16)-(17). Then*

(i) $D_j \leq 0$ deterministically, with no assumption on μ, Σ, e ;

(ii) $|\mathcal{E}_j| \leq \frac{s_j^2}{2} |S_j| \rho(\Sigma)$.

Proof. (i) *Sign of D_j .* Each summand $-s_j \Sigma_{ii} |e_{i,j}| \leq 0$, since $s_j > 0$, $\Sigma_{ii} \geq 0$ (PSD diagonal), and $|e_{i,j}| \geq 0$.

(ii) *Bound on $|\mathcal{E}_j|$.* By the triangle inequality and the nearest-level residual bound $|e_{k,j}| \leq s_j/2$ (Section 3.1) (which holds for every $k \in \{1, \dots, d\}$ under plain RTN with Assumption 2; this is the property that breaks under GPTQ's sequential updates and AdaRound's learned rounding, see Appendix G),

$$\begin{aligned} |\mathcal{E}_j| &\leq s_j \sum_{i \in S_j} \sum_{k \neq i} |\Sigma_{ik}| |e_{k,j}| \\ &\leq \frac{s_j^2}{2} \sum_{i \in S_j} \sum_{k \neq i} |\Sigma_{ik}| \leq \frac{s_j^2}{2} |S_j| \rho(\Sigma). \end{aligned}$$

Summing (16) over channels, $|D_\ell| = \sum_{(i,j): i \in S_j} s_j \Sigma_{ii} |e_{i,j}|$ is a sum of K_ℓ terms each $\geq \gamma_S$, so

$$|D_\ell| \geq K_\ell \gamma_S. \quad (18)$$

Summing the lemma's off-diagonal bound,

$$|\mathcal{E}_\ell| \leq \sum_j \frac{s_j^2}{2} |S_j| \rho(\Sigma) \leq \frac{s_{\max}^2}{2} K_\ell \rho(\Sigma). \quad (19)$$

C.3 Stage 3: the trace term and assembly

By Assumption 1, Δ_j is supported on $S_j \subseteq \mathcal{I}$, so the quadratic form sees only the principal submatrix $\Sigma_{\mathcal{II}}$: $\Delta_j^\top \Sigma \Delta_j = \Delta_j^\top \Sigma_{\mathcal{II}} \Delta_j \leq \|\Sigma_{\mathcal{II}}\|_2 \|\Delta_j\|_2^2$. The unit-flip rule (Section 3.2) fixes $\|\Delta_j\|_2^2 = s_j^2 |S_j|$, hence

$$\begin{aligned} \text{tr}(\Delta^\top \Sigma \Delta) &\leq \|\Sigma|_{\mathcal{I}}\|_2 \sum_j s_j^2 |S_j| \\ &\leq s_{\max}^2 K_\ell \|\Sigma|_{\mathcal{I}}\|_2. \end{aligned} \quad (20)$$

Proof of Theorem 1. From (15) and Lemma 1, $\Delta \mathcal{R}_\ell = -G_\ell + 2(D_\ell + \mathcal{E}_\ell) + \text{tr}(\Delta^\top \Sigma \Delta)$. Using $D_\ell \leq -|D_\ell|$ and $\mathcal{E}_\ell \leq |\mathcal{E}_\ell|$, then (18), (19), (20),

$$\begin{aligned} \Delta \mathcal{R}_\ell &\leq -G_\ell - 2|D_\ell| + 2|\mathcal{E}_\ell| + \text{tr}(\Delta^\top \Sigma \Delta) \\ &\leq -G_\ell - 2K_\ell \gamma_S + s_{\max}^2 K_\ell \rho(\Sigma) \\ &\quad + s_{\max}^2 K_\ell \|\Sigma|_{\mathcal{I}}\|_2 \\ &= -G_\ell + K_\ell \left[-2\gamma_S + s_{\max}^2 (\rho(\Sigma) + \|\Sigma|_{\mathcal{I}}\|_2) \right] \\ &= -G_\ell + K_\ell \Phi_{\mathcal{I}}, \end{aligned}$$

which is (6) with $\Phi_{\mathcal{I}}$ as stated.

Remark (looseness is by design). The bound treats two quantities adversarially: $\rho(\Sigma)$ assumes every off-diagonal entry of Σ is aligned with e , and $\|\Sigma|_{\mathcal{I}}\|_2$ assumes Δ_j points along the top eigenvector of $\Sigma|_{\mathcal{I}}$. Neither holds in practice, which is why the realized variance penalty in Appendix E.1 is orders of magnitude below $K_\ell \Phi_{\mathcal{I}}$. The deterministic piece $D_\ell \leq 0$ (Lemma 1) is the mechanism that actually drives the variance change negative on real layers; the worst-case $\Phi_{\mathcal{I}}$ only certifies that the change cannot blow up.

D Proof of Corollary 1

Proof plan. Theorem 1 gives $\Delta \mathcal{R}_\ell \leq -G_\ell + K_\ell \Phi_{\mathcal{I}}$. We simply ask, in each sign regime of $\Phi_{\mathcal{I}}$, for which budgets K_ℓ this right-hand side is ≤ 0 . The two cases are immediate.

Proof. Regime (i). If $\Phi_{\mathcal{I}} \leq 0$, write $\Phi_{\mathcal{I}} = -|\Phi_{\mathcal{I}}|$; then $\Delta \mathcal{R}_\ell \leq -G_\ell - K_\ell |\Phi_{\mathcal{I}}| \leq 0$ for any $K_\ell \geq 0$, strictly so whenever $G_\ell > 0$ or ($K_\ell > 0$ and $|\Phi_{\mathcal{I}}| > 0$).

Regime (ii). If $\Phi_{\mathcal{I}} > 0$, the bound $-G_\ell + K_\ell \Phi_{\mathcal{I}}$ is ≤ 0 iff $K_\ell \Phi_{\mathcal{I}} \leq G_\ell$, i.e. $K_\ell \leq G_\ell / \Phi_{\mathcal{I}}$; the inequality is strict for $K_\ell < G_\ell / \Phi_{\mathcal{I}}$. In either regime the set of budgets certifying $\Delta \mathcal{R}_\ell \leq 0$ is non-empty.

E Bias Descent of the Greedy Rule

Proof plan. We show the per-channel selection (8) can never increase b_j^2 . The selection minimizes $|b_j^{(k)}|$ over a feasible range that includes $k = 0$ (the empty selection, value $|b_j^{(0)}| = |b_j|$); optimality over a set containing $k = 0$ forces $|b_j'| \leq |b_j|$. Summing over channels gives $G_\ell \geq 0$.

Proposition 1. *For each channel j , the greedy rule (8) satisfies $b_j'^2 \leq b_j^2$, where $b_j' := b_j^{(k^*)}$. Consequently $G_\ell \geq 0$.*

Proof. Recall $b_j^{(k)} := b_j - \sum_{t \leq k} v_{it,j}$ and the selection $k^* = \arg \min_{0 \leq k \leq B_j} |b_j^{(k)}|$. The feasible range includes $k = 0$, the empty selection, whose objective value is $|b_j^{(0)}| = |b_j|$. Since k^* is optimal over a set containing $k = 0$, $|b_j'| = |b_j^{(k^*)}| \leq |b_j|$, hence $\mathcal{B}_j(W_q') = b_j'^2 \leq b_j^2 = \mathcal{B}_j(W_q)$, where $\mathcal{B}_j := b_j^2$ is the per-channel bias term of (2). Summing over j gives $G_\ell = \mathcal{B}_\ell(W_q) - \mathcal{B}_\ell(W_q') \geq 0$.

Table 3: Empirical verification of every component of the CLC theory on Mistral-7B-v0.3 with plain RTN base (16 layers per setting). Each check is reported as the fraction of layers on which it passes. The exact identity (5), the one-level correction rule and Assumptions 1-2, and the bias descent hold on every layer. The on-support diagonal D_ℓ is non-positive on every layer (Lemma 1(i)), and the off-diagonal bound (Lemma 1(ii)) and the trace bound both hold with substantial slack. The master bound of Theorem 1 holds universally. Layer error descends on every measured layer (universal empirical descent).

Check	in-sample	cross-eval
Eq. (5) identity holds (rel. resid. $< 10^{-13}$)	16/16	16/16
One-level rule: $\Delta_{i,j} = -\text{sign}(e_{i,j}) s_j$	16/16	16/16
Assumption 1: $\text{supp}(\Delta_j) \subseteq \mathcal{I}$ (minimal $\mathcal{I}$)	16/16	16/16
Assumption 2: uniform per-channel step s_j	16/16	16/16
Bias descent: $G_\ell \geq 0$	16/16	16/16
Lemma 1: $\langle \Delta, \Sigma e \rangle = D + \mathcal{E}$ exactly	16/16	16/16
Lemma 1(i): $D_\ell \leq 0$ (deterministic)	16/16	16/16
Lemma 1(ii): $ \mathcal{E}_j \leq \frac{s_j^2}{2} S_j \rho(\Sigma)$ per row	16/16	16/16
$ D_\ell \geq K_\ell \gamma_S$	16/16	16/16
Trace bound: $\text{tr}(\Delta^\top \Sigma \Delta) \leq s_{\max}^2 K_\ell \ \Sigma_{\mathcal{I}\mathcal{I}}\ _2$	16/16	16/16
Theorem 1 master bound: $\Delta \mathcal{R}_\ell \leq -G_\ell + K_\ell \Phi_{\mathcal{I}}$	16/16	16/16
Realized layer descent $\Delta \mathcal{R}_\ell < 0$	16/16	16/16

Table 4: Realized magnitudes (layer-averaged) on Mistral-7B-v0.3 with plain RTN base. The on-support diagonal D_ℓ accounts for the bulk of the cross-term and is the dominant mechanism of descent. The off-diagonal residual $\mathcal{E}_\ell$ is itself empirically negative, contributing additional descent. The bound’s $\Phi_{\mathcal{I}}$ is positive on every layer (regime (ii) of Corollary 1), so the master bound certifies descent only via the safety budget $K_\ell < G_\ell / \Phi_{\mathcal{I}}$; in practice the realized cross-term and trace are much smaller than the worst-case estimate, and descent is universal. Aggregate $\mathcal{R}$ across the 16 layers drops by 15.7% in-sample and 15.9% cross-eval.

Quantity	in-sample	cross-eval
$\bar{D}_\ell$	-9.39×10^{-2}	-9.39×10^{-2}
$\bar{\mathcal{E}}_\ell$	-3.06×10^{-2}	-3.06×10^{-2}
$\langle \Delta, \Sigma e \rangle$	-1.25×10^{-1}	-1.25×10^{-1}
$\text{tr}(\Delta^\top \Sigma \Delta)$	$+2.17 \times 10^{-1}$	$+2.17 \times 10^{-1}$
$\bar{\mathcal{E}}_\ell$ bound (Lemma 1(ii))	$+7.33 \times 10^3$	$+7.33 \times 10^3$
trace bound (RHS)	$+1.72 \times 10^4$	$+1.72 \times 10^4$
$\bar{\Phi}_{\mathcal{I}}$	$+8.72 \times 10^{-1}$	$+8.72 \times 10^{-1}$
$\bar{\Delta \mathcal{R}}_\ell$ (realized)	-5.03×10^{-1}	-5.39×10^{-1}
Aggregate $\mathcal{R}$ change	-15.7%	-15.9%

E.1 Empirical Verification of the Theory

We verify every component of the analysis in §3.2-3.3 directly - the exact identity (5), the one-level correction rule and the algorithmic assumptions (Assumptions 1-2), the bias descent (Proposition 1), Lemma 1, the trace bound, and the master bound of Theorem 1 - by instrumenting 16 linear layers (uniformly sampled across attention and MLP projections) of Mistral-7B-v0.3 quantized with plain RTN at 4-bit weight-only, group size 128. We compute $\hat{\mu}_\ell$ via James-Stein (7) and the full covariance Σ_ℓ from 128 C4 calibration samples, apply CLC with budget $p = 5\%$, and measure every component of the bound. Results are reported under two regimes: *in-sample* (statistics and evaluation on C4) and *cross-eval* (statistics from C4, evaluation

on WikiText-2), the latter probing calibration drift. Throughout, K_ℓ denotes the realized number of flips, D_ℓ the on-support diagonal piece of the cross-term in Lemma 1, and $\mathcal{E}_\ell := \langle \Delta_\ell, \Sigma_\ell e_\ell \rangle - D_\ell$ the off-diagonal residual.

Tables 3 and 4 summarize the verification. The measurements confirm that every step of the analysis holds on real layers, and they make explicit where the worst-case bound is loose versus where the structural mechanism actually drives descent.

Verification 1: the exact identity and the algorithmic assumptions. Equation (5) holds to floating-point precision on every layer (relative residual below 10^{-13}), confirming that the $-G_\ell + \Delta \mathcal{V}_\ell$ split is exact rather than approximate. The three structural conditions on Δ - the one-level cor-

rection rule of Section 3.2 (anti-aligned unit flip), Assumption 1 (common support, verified for the minimal $\mathcal{I} = \bigcup_j S_j$), and Assumption 2 (uniform per-channel step) - and the bias descent ($G_\ell \geq 0$, Proposition 1) are each satisfied on every layer.

Verification 2: structural non-positivity of D_ℓ . On every layer (16/16 in-sample and 16/16 cross-eval), the deterministic on-support diagonal D_ℓ is non-positive, confirming Lemma 1(i) and validating that the anti-aligned flip rule of Section 3.2 delivers a strictly non-positive variance-side contribution *without any statistical assumption on the activation distribution*. This is the structural guarantee the algorithm was designed to extract: Table 4 shows $|D_\ell|$ is on average roughly three times $|\mathcal{E}_\ell|$, so the diagonal piece dominates the cross-term in practice.

Verification 3: bounds hold but are loose. The off-diagonal bound of Lemma 1(ii), the lower bound on $|D_\ell|$, the trace bound, and the master bound of Theorem 1 all hold on every layer. They are, however, very loose. Comparing realized values to bounds in Table 4: $|\mathcal{E}_\ell|$ is roughly five orders of magnitude smaller than its bound; the realized trace is five orders of magnitude smaller than the trace bound; the realized $\Delta\mathcal{R}_\ell$ is a small negative number while the right-hand side $-G_\ell + K_\ell\Phi_{\mathcal{I}}$ of (6) is large and positive. The looseness has two sources: $\rho(\Sigma)$ is a worst-case row-sum that treats every off-diagonal entry as adversarially aligned with e , and $\|\Sigma|_{\mathcal{I}}\|_2$ treats Δ_j as adversarially aligned with the top eigenvector of $\Sigma|_{\mathcal{I}}$. Neither alignment occurs in practice. We do not view this as a defect: the bound is a worst-case certificate, and the realized values in Table 4 are what governs CLC’s behavior.

Verification 4: empirical regime versus theoretical regime. A consequence of the loose bound is that $\Phi_{\mathcal{I}}$ is positive on every measured layer (mean $\bar{\Phi}_{\mathcal{I}} \approx +0.87$), placing each layer in regime (ii) of Corollary 1, where the bound’s guarantee of descent is *conditional* on the safety budget $K_\ell \leq G_\ell/\Phi_{\mathcal{I}}$. Yet *realized* descent is universal: $\Delta\mathcal{R}_\ell < 0$ on every layer in both in-sample and cross-eval, with aggregate $\mathcal{R}$ dropping by $\sim 16\%$. The mechanism is exactly the structural one identified in Lemma 1(i): D_ℓ is reliably negative and large in magnitude, the realized $\mathcal{E}_\ell$ is also negative in practice (its sign is not guaranteed by the lemma but is consistent across layers), and the realized

trace is positive but small. The bound’s regime classification is therefore conservative; descent in practice is the rule, not a conditional event.

Verification 5: robustness to calibration drift.

The cross-eval results - calibration μ estimated on C4, evaluation on WikiText-2 - show no qualitative change: every check still passes on every layer and descent remains universal. The mean realized $\Delta\mathcal{R}_\ell$ shifts only slightly (from -0.503 to -0.539), because the bias term $b_j^2 = (\mu^\top e_j)^2$ is evaluated against a different μ and the descent shrinks accordingly, but the mechanism is intact. James-Stein shrinkage on $\hat{\mu}$ is what keeps the support-selection step stable across datasets; without it the knee threshold τ would move with calibration noise.

Implications for hyperparameter choice. The empirical placement on the optimistic side of regime (ii) - the bound sits in regime (ii), but realized descent is universal - has a concrete practical implication. The budget p acts as a magnitude knob because the realized cross-term-plus-trace stays much smaller than the worst-case $K_\ell\Phi_{\mathcal{I}}$, so descent grows roughly with G_ℓ rather than reversing as K_ℓ grows. Diminishing returns at larger p in the ablations therefore reflect saturation of G_ℓ once the per-channel bias residual b_j is brought close to zero, not the variance-cost crossover the bound conservatively predicts. The bottleneck in CLC’s empirical regime is the bias-gain term, not the variance-cost term - exactly the inversion of bias correction’s limitation.

F Algorithm of Complementary Lattice Correction

Algorithm 1 summarizes the constructive procedure described in Section 3.4. CLC first forms a stabilized estimate of the activation mean and uses it to define a common admissible support $\mathcal{I}$. For each output channel, the algorithm then constructs anti-residual adjacent-level moves, retains only candidates aligned with the channel mean-shift direction, and ranks them by residual magnitude. The final corrected weights are obtained by applying the budgeted prefix that minimizes the calibration mean shift for each channel.

G How CLC Interfaces with Existing Quantization Pipelines

The applicability of CLC depends on the discrete representation exposed after a base quantization

Algorithm 1 Complementary Lattice Correction (CLC)

Input: RTN-quantized weights $W_q \in \mathbb{R}^{d \times C}$; full-precision weights W ; calibration activations $X \in \mathbb{R}^{d \times m}$; per-channel steps $\{s_j\}_{j=1}^C$; flip budget fraction $p \in (0, 1]$.

Output: Corrected quantized weights W'_q .

```

1: // Estimate activation mean
2:  $\bar{x} \leftarrow \frac{1}{m} X \mathbf{1}_m$ 
3:  $\hat{\mu} \leftarrow \text{JamesStein}(\bar{x})$  via Eq. (7)
4: // Support mask (proxy for  $\mathcal{I}$ )
5:  $a_i \leftarrow |\hat{\mu}_i|$  for  $i = 1, \dots, d$ ; sort descending
6:  $\tau \leftarrow \text{Kneedle threshold on } \{a_{(t)}\}_{t=1}^{\lfloor d/2 \rfloor}$ 
7:  $\mathcal{I} \leftarrow \{i : |\hat{\mu}_i| \leq \tau\}$ 
8:  $W'_q \leftarrow W_q$ 
9: for  $j = 1, \dots, C$  do
10:   $e_j \leftarrow w_{q, :, j} - w_{:, j}$ ;  $b_j \leftarrow \hat{\mu}^\top e_j$ 
11:  for  $i \in \mathcal{I}$  do
12:     $\sigma_{i,j} \leftarrow -\text{sign}(e_{i,j})$ 
13:     $v_{i,j} \leftarrow -\hat{\mu}_i \cdot \sigma_{i,j} \cdot s_j$ 
14:  end for
15:   $\mathcal{I}_j \leftarrow \{i \in \mathcal{I} : \text{sign}(v_{i,j}) = \text{sign}(b_j)\}$ 
16:   $B_j \leftarrow \lceil p \cdot |\mathcal{I}_j| \rceil$ 
17:   $n_j \leftarrow |\mathcal{I}_j|$ 
18:  Order  $\mathcal{I}_j$  by descending residual magnitude
     $\delta_{i,j} := |e_{i,j}| : (i_1, \dots, i_{n_j})$ 
19:   $k^* \leftarrow \arg \min_{0 \leq k \leq B_j} |b_j - \sum_{t=1}^k v_{i_t, j}|$ 
20:   $S_j \leftarrow \{i_1, \dots, i_{k^*}\}$ 
21:  for  $i \in S_j$  do
22:     $w'_{q, i, j} \leftarrow w_{q, i, j} + \sigma_{i,j} \cdot s_j$ 
23:  end for
24: end for
25: return  $W'_q$ 

```

pipeline has finished. The analysis in §3.3 assumes that, within each output channel, the stored weights lie on a uniform lattice governed by a single step s_j (Assumption 2), so that a correction can be expressed as an adjacent-level reassignment on that lattice. This condition is immediate for the baseline setting analyzed in the main text. For stronger PTQ pipelines, however, the same lattice representation is often reached only after method-specific transformations, updates, or learned assignment rules. These preceding operations do not necessarily prevent CLC from being applied, but they change the coordinate system or residual structure in which the CLC rule should be interpreted.

We therefore make the interface between CLC and several representative pipelines explicit. In

the order AWQ, GPTQ, FlatQuant and AdaRound, we identify (a) the transformation or optimization performed before the final lattice assignment, (b) the point at which nearest-level or nearest-level-related discretization produces the stored integer codes, and (c) the additional issue that must be handled before the CLC analysis can be transferred directly.

Notation. Throughout, $W \in \mathbb{R}^{d \times C}$ is the full-precision weight matrix of a single linear layer with d input dimensions and C output channels, $X \in \mathbb{R}^{d \times m}$ collects calibration activations, and $\text{RTN}_s(\cdot)$ denotes the per-channel asymmetric round-to-nearest projection onto the uniform integer lattice with step s .

G.1 AWQ: input-channel pre-scaling

AWQ (Lin et al., 2024) rescales the layer before quantization using activation-aware input-channel factors. Under our convention $W \in \mathbb{R}^{d \times C}$ and layer output $W^\top X$, let $S = \text{diag}(\alpha_1, \dots, \alpha_d)$ denote the diagonal scaling matrix. The equivalent transformed layer is

$$\begin{aligned} W^\top X &= (SW)^\top (S^{-1}X) \\ &\approx Q(SW)^\top (S^{-1}X), \end{aligned} \quad (21)$$

where $Q(\cdot)$ denotes the final nearest-level lattice assignment. Thus, the discretized object is the scaled weight matrix SW , and the activation statistics relevant to the transformed layer are those of $S^{-1}X$.

In this coordinate system, AWQ is relatively compatible with CLC. If the scaled matrix is quantized with a single lattice step per output channel, Assumption 2 is preserved in the transformed coordinates. The residual corrected by CLC is then

$$e' = Q(SW) - SW,$$

and its first-moment alignment should be computed against the transformed activation mean

$$\mu' = S^{-1}\mu, \quad \Sigma' = S^{-1}\Sigma S^{-1}.$$

Therefore, the CLC analysis can be applied by replacing (W, μ, Σ) with (SW, μ', Σ') .

The main change is in the constants of the bound: the support covariance and off-diagonal interaction terms are evaluated under the transformed covariance Σ' , rather than the original Σ . Thus AWQ does not fundamentally change the adjacent-level correction rule, but it changes the coordinate system in which the correction and its worst-case variance cost should be analyzed.

G.2 GPTQ: second-order updates before lattice assignment

GPTQ (Frantar et al., 2023) quantizes weights sequentially along the input-column dimension. Let $H \approx XX^\top + \lambda I$ denote the damped Hessian estimate formed from calibration activations. After a column has been quantized, GPTQ uses entries of H^{-1} to propagate its quantization error into the remaining floating-point columns. Thus, when column k is discretized, the value assigned to the lattice is not necessarily the original weight $w_{j,k}$, but a GPTQ-updated pre-round value

$$\tilde{w}_{j,k} = w_{j,k} + c_{j,k}^{\text{GPTQ}},$$

where $c_{j,k}^{\text{GPTQ}}$ denotes the accumulated Hessian-aware correction from previously quantized columns. The corresponding integer code and stored weight are

$$q_{j,k} = \text{round}\left(\frac{\tilde{w}_{j,k}}{s_j} + z_j\right),$$

$$w_{q,j,k} = s_j(q_{j,k} - z_j).$$

Crucially, GPTQ modifies the pre-rounding weights but does *not* change the quantization lattice, so CLC still operates on the same set of integer levels.

Objective and admissible move. Our CLC implementation uses the original weight and the GPTQ pre-round value for different roles. The correction objective is measured against the original full-precision layer,

$$\hat{b}_j = \hat{\mu}^\top(w_{q,j} - w_j),$$

because this is the first-moment error seen by the final quantized model. In contrast, the adjacent-level direction is determined from GPTQ’s local pre-round state. In normalized coordinates, CLC uses the rounding residual

$$r_{j,k} = \frac{\tilde{w}_{j,k}}{s_j} + z_j - q_{j,k},$$

and proposes the one-level move

$$q'_{j,k} = q_{j,k} + \text{sign}(r_{j,k}),$$

whenever the proposed code remains within the valid quantization range. The same quantity $|r_{j,k}|$ is used to rank candidate flips.

Which guarantees carry over. The first-moment descent guarantee remains valid under GPTQ: because the greedy prefix search includes the no-flip option, the selected update can never increase the estimated channel-wise first-moment error,

$$|\hat{b}'_j| \leq |\hat{b}_j|.$$

The worst-case MSE guarantee of Theorem 1, however, does *not* apply verbatim. GPTQ’s compensated weights may violate the RTN residual condition

$$|e_{j,k}| \leq \frac{s_j}{2}$$

that the bound assumes, and more fundamentally the separation of roles above differs from the plain nearest-level setting, where a single residual both defines the flip direction and appears in the CLC error decomposition. In other words, under GPTQ the first-moment descent is preserved but the certified non-increasing-MSE condition is no longer guaranteed to hold. In practice we monitor the effect of CLC through GPTQ’s layer-wise reconstruction objective $\|W_q X - W X\|_F^2$ and observe that the additional adjacent-level flips leave it essentially unchanged while reducing the first-moment error.

Thus, after GPTQ, CLC acts as a lattice refinement step: GPTQ and H^{-1} determine the local pre-round geometry, while CLC selects adjacent-level changes according to their effect on the original layer-wise mean-shift error. A complete theory would require a GPTQ-aware analysis of the interaction between second-order updates, rounding-level choices, and inference-time activation statistics; we leave such rounding-selection rules to future work.

G.3 FlatQuant: learned transformed coordinates before lattice assignment

FlatQuant (Sun et al., 2025) first calibrates and absorbs learned transformations into the model before weight quantization. These transformations may include Hadamard-like or Kronecker-structured rotations, learned diagonal rescaling, and additional weight/activation calibration parameters. Under our convention $W \in \mathbb{R}^{d \times C}$ and layer output $W^\top X$, we abstract the resulting reparameterized linear layer as

$$W^\top X = (W')^\top X',$$

where W' is the post-transformation weight matrix and X' is the corresponding transformed activation.

The final quantized weights are then obtained by assigning W' to integer lattice levels,

$$W'_q = Q(W').$$

In our implementation, CLC is applied after this reparameterization. Therefore, the correction is performed directly in the transformed coordinate system: the quantization state is built from the post-FlatQuant weight W' , and the activation mean is estimated from the inputs entering the reparameterized linear layer. The mean-shift term is thus

$$\hat{b}'_j = (\hat{\mu}')^\top (w'_{q,j} - w'_j),$$

where $\hat{\mu}'$ is computed from X' . The adjacent-level move is the same one-level lattice correction used in the base CLC procedure, but applied to the lattice of W'_q .

From the perspective of CLC, FlatQuant is therefore similar to AWQ: the base method changes the quantization coordinate system before discretization, and CLC operates on the lattice exposed in that coordinate system. This view suggests a broader interface for transformation-based quantization methods: once the transformation has been absorbed into the model, CLC can be applied as a generic post-quantization lattice refinement using the weights and activation statistics in the transformed space.

G.4 AdaRound: learned per-weight rounding decisions

AdaRound (Nagel et al., 2020) replaces deterministic nearest-level rounding with learned up/down decisions. For a normalized weight $u_{j,k} = w_{j,k}/s_j + z_j$, the stored integer code can be written as

$$q_{j,k} = \lfloor u_{j,k} \rfloor + h_{j,k}, \quad h_{j,k} \in \{0, 1\},$$

with $h_{j,k}$ obtained by optimizing a continuous relaxation under a local reconstruction objective. The stored weight is $w_{q,j,k} = s_j(q_{j,k} - z_j)$.

The resulting weights still lie on the same per-channel integer lattice, so the lattice-step condition in Assumption 2 is preserved. The difficulty is that the local rounding decision no longer follows the nearest-level rule. AdaRound may deliberately choose the farther of the two neighboring lattice levels if that improves its reconstruction objective. Consequently, the residual $w_{q,j,k} - w_{j,k}$ need not be bounded by $s_j/2$, and its sign no longer carries the same meaning as in plain nearest-level quantization.

For CLC, the natural adjacent move after AdaRound is therefore not “move toward the nearest level”, but rather reopen the learned binary choice:

$$q'_{j,k} = \lfloor u_{j,k} \rfloor + (1 - h_{j,k}).$$

The mean-shift objective can still be evaluated against the original full-precision layer, but each candidate flip must now be interpreted as reversing one of AdaRound’s learned rounding decisions. Thus, the gain predicted by the CLC first-moment criterion must be balanced against the fact that AdaRound has already optimized these decisions for a local reconstruction loss.

Therefore, AdaRound exposes a valid lattice for CLC, but the clean CLC assumptions do not transfer directly: the residual used in the original nearest-level analysis is no longer the same object that defines the learned rounding assignment. A complete extension would require AdaRound-aware rules for deciding which learned rounding choices should be reopened for first-moment correction, which we leave to future work.

H Additional Experimental Results

Experimental Setup. All experiments are conducted with Python3.12, using PyTorch¹ and the Transformers library² to deploy large language models. All LLM checkpoints are obtained from their official Hugging Face repositories³. Due to the high memory and bandwidth requirements of LLM inference and quantization, we run all experiments on NVIDIA A100 GPUs. Zero-shot downstream-task evaluation is carried out using the lm-evaluation-harness tool⁴. For LLaMA-3-8B with FlatQuant⁵, we use a separate environment following the original FlatQuant implementation, since its LLaMA-3 transformation pipeline relies on earlier versions of several dependencies.

Overview. The main paper keeps the experimental section compact by reporting the most representative settings, but the central claim of CLC is broader: the correction should not depend on a particular model family, a single bit-width, or a specific PTQ backend. This appendix therefore

¹<https://pytorch.org/>

²<https://huggingface.co/docs/transformers>

³<https://huggingface.co/>

⁴<https://github.com/EleutherAI/>

lm-evaluation-harness

⁵<https://github.com/ruikangliu/FlatQuant>

Table 5: 3-bit quantization results on Qwen2.5-7B and LLaMA-3-8B. PPL on WikiText-2 and C4 ($\downarrow$); zero-shot accuracy on five tasks and their average ($\uparrow$). Bold marks the best non-FP16 row within each (model, base) block. Green shading marks the best non-FP16 result within each base-quantizer block, while red shading marks metrics degraded relative to the corresponding original quantizer.

Model	Method	PPL (↓)		Downstream Accuracy (↑)						
		Wiki	C4	arc-c	arc-e	boolq	piqa	rte	Avg	
Qwen2.5-7B	FP16	6.084 ±0.004	10.817 ±0.007	0.504 ±0.002	0.804 ±0.003	0.846 ±0.001	0.787 ±0.003	0.835 ±0.002	0.755	
	RTN	10.793 ±0.010	17.638 ±0.021	0.398 ±0.004	0.701 ±0.001	0.747 ±0.003	0.746 ±0.001	0.663 ±0.003	0.651	
	RTN + BC	10.669 ±0.014	17.587 ±0.015	0.393 ±0.003	0.701 ±0.001	0.753 ±0.003	0.747 ±0.002	0.652 ±0.003	0.649	
	RTN + CLC	8.971 ±0.009	15.237 ±0.012	0.417 ±0.003	0.726 ±0.003	0.768 ±0.002	0.763 ±0.001	0.666 ±0.003	0.668	
	AWQ	8.592 ±0.008	15.048 ±0.014	0.439 ±0.001	0.731 ±0.002	0.784 ±0.003	0.769 ±0.003	0.678 ±0.001	0.680	
	AWQ + BC	8.609 ±0.010	15.012 ±0.017	0.434 ±0.002	0.728 ±0.001	0.763 ±0.002	0.770 ±0.003	0.614 ±0.001	0.662	
	AWQ + CLC	8.231 ±0.009	14.243 ±0.010	0.441 ±0.003	0.729 ±0.004	0.812 ±0.003	0.779 ±0.003	0.690 ±0.003	0.690	
	GPTQ	8.138 ±0.006	15.010 ±0.009	0.423 ±0.002	0.724 ±0.002	0.782 ±0.003	0.769 ±0.002	0.670 ±0.003	0.674	
	GPTQ + BC	8.147 ±0.004	15.017 ±0.011	0.418 ±0.002	0.724 ±0.002	0.782 ±0.004	0.768 ±0.001	0.678 ±0.004	0.674	
	GPTQ + CLC	8.090 ±0.003	14.994 ±0.009	0.428 ±0.002	0.719 ±0.004	0.784 ±0.003	0.776 ±0.002	0.685 ±0.002	0.678	
	FlatQuant	7.257 ±0.006	12.415 ±0.011	0.475 ±0.002	0.795 ±0.002	0.823 ±0.002	0.776 ±0.001	0.758 ±0.004	0.725	
	FlatQuant + BC	7.247 ±0.005	12.401 ±0.009	0.458 ±0.003	0.770 ±0.002	0.834 ±0.002	0.776 ±0.003	0.765 ±0.002	0.721	
FlatQuant + CLC	7.221 ±0.006	12.401 ±0.008	0.502 ±0.001	0.803 ±0.001	0.827 ±0.002	0.777 ±0.003	0.773 ±0.001	0.736		
LLaMA-3-8B	FP16	5.442 ±0.004	8.346 ±0.007	0.505 ±0.001	0.800 ±0.002	0.813 ±0.001	0.796 ±0.004	0.696 ±0.003	0.722	
	RTN	10.947 ±0.011	18.044 ±0.010	0.386 ±0.004	0.649 ±0.003	0.717 ±0.004	0.743 ±0.003	0.588 ±0.003	0.617	
	RTN + BC	10.502 ±0.005	17.742 ±0.020	0.349 ±0.003	0.649 ±0.002	0.688 ±0.001	0.714 ±0.004	0.560 ±0.002	0.592	
	RTN + CLC	9.105 ±0.009	15.251 ±0.016	0.406 ±0.002	0.670 ±0.002	0.707 ±0.002	0.751 ±0.003	0.611 ±0.003	0.629	
	AWQ	8.136 ±0.008	13.293 ±0.004	0.394 ±0.002	0.697 ±0.002	0.744 ±0.001	0.759 ±0.001	0.613 ±0.002	0.641	
	AWQ + BC	8.140 ±0.005	13.310 ±0.009	0.400 ±0.002	0.700 ±0.004	0.742 ±0.002	0.759 ±0.002	0.635 ±0.004	0.647	
	AWQ + CLC	7.973 ±0.002	12.977 ±0.007	0.403 ±0.001	0.707 ±0.003	0.750 ±0.003	0.766 ±0.003	0.614 ±0.002	0.648	
	GPTQ	9.404 ±0.005	12.649 ±0.006	0.373 ±0.004	0.700 ±0.003	0.717 ±0.002	0.756 ±0.004	0.620 ±0.002	0.633	
	GPTQ + BC	9.407 ±0.008	12.647 ±0.004	0.373 ±0.003	0.699 ±0.003	0.716 ±0.004	0.755 ±0.002	0.620 ±0.003	0.633	
	GPTQ + CLC	9.403 ±0.003	12.642 ±0.009	0.374 ±0.002	0.702 ±0.001	0.718 ±0.002	0.759 ±0.001	0.625 ±0.003	0.636	
	FlatQuant	7.451 ±0.006	11.980 ±0.003	0.417 ±0.004	0.749 ±0.003	0.791 ±0.002	0.768 ±0.003	0.625 ±0.002	0.670	
	FlatQuant + BC	7.424 ±0.010	12.052 ±0.005	0.427 ±0.001	0.730 ±0.003	0.769 ±0.001	0.755 ±0.001	0.585 ±0.003	0.653	
FlatQuant + CLC	7.408 ±0.007	11.933 ±0.008	0.425 ±0.003	0.750 ±0.002	0.781 ±0.003	0.763 ±0.002	0.646 ±0.001	0.673		

evaluates the same post-processing rule along three complementary axes. First, we add two LLMs, Qwen2.5-7B and LLaMA-3-8B, to test whether the trend observed on the main models transfers across architectures and pretraining recipes. Second, we report both 4-bit and 3-bit results, where 3-bit quantization provides a substantially harder stress test because the lattice spacing is larger and projection residuals become more structured. Third, we include AdaRound, a learned-rounding quantizer, to test whether CLC remains useful even when the base method has already optimized rounding decisions using calibration data.

H.1 Additional Models at 3-bit

Table 5 reports the additional 3-bit results on Qwen2.5-7B and LLaMA-3-8B. This is a stringent setting for post-quantization correction because the coarser integer lattice introduces larger projection residuals, making structured first-moment errors more visible after the base quantizer has finished. Across all eight model-quantizer blocks, CLC improves both perplexity metrics and the un-weighted average downstream accuracy over the

corresponding base quantizer, although individual task accuracies can still fluctuate. This provides strong additional evidence that CLC is not specific to the two main-paper models and that its effect becomes more pronounced as the lattice resolution decreases.

The largest gains appear for RTN, where the final nearest-level projection is least protected by additional reconstruction or transformation steps. On Qwen2.5-7B, CLC reduces RTN WikiText-2 PPL from 10.793 to 8.971 and C4 PPL from 17.638 to 15.237, while improving the average zero-shot score from 0.651 to 0.668. On LLaMA-3-8B, CLC similarly reduces RTN WikiText-2 PPL from 10.947 to 9.105 and C4 PPL from 18.044 to 15.251, with the average accuracy increasing from 0.617 to 0.629. These are large gains for a post-processing step that does not retrain the model, reconstruct layers with SGD, or alter the inference graph.

CLC also remains effective on stronger 3-bit quantizers. With AWQ, it improves Qwen2.5-7B from 8.592 to 8.231 WikiText-2 PPL and from 15.048 to 14.243 C4 PPL, while raising the average

Table 6: 4-bit quantization results on Qwen2.5-7B and LLaMA-3-8B. PPL on WikiText-2 and C4 ($\downarrow$); zero-shot accuracy on five tasks and their average ($\uparrow$). Bold marks the best non-FP16 row within each (model, base) block. Green shading marks the best non-FP16 result within each base-quantizer block, while red shading marks metrics degraded relative to the corresponding original quantizer.

Model	Method	PPL ($\downarrow$)		Downstream Accuracy ($\uparrow$)					
		Wiki	C4	arc-c	arc-e	boolq	piqa	rte	Avg
Qwen2.5-7B	FP16	6.084 $\pm$ 0.004	10.817 $\pm$ 0.007	0.504 $\pm$ 0.002	0.804 $\pm$ 0.003	0.846 $\pm$ 0.001	0.787 $\pm$ 0.003	0.835 $\pm$ 0.002	0.755
	RTN	6.692 $\pm$ 0.006	12.203 $\pm$ 0.003	0.451 $\pm$ 0.003	0.778 $\pm$ 0.003	0.830 $\pm$ 0.004	0.766 $\pm$ 0.001	0.802 $\pm$ 0.002	0.725
	RTN + BC	6.643 $\pm$ 0.005	12.108 $\pm$ 0.004	0.441 $\pm$ 0.004	0.768 $\pm$ 0.001	0.827 $\pm$ 0.004	0.765 $\pm$ 0.004	0.823 $\pm$ 0.003	0.725
	RTN + CLC	6.637 $\pm$ 0.007	11.878 $\pm$ 0.002	0.452 $\pm$ 0.002	0.771 $\pm$ 0.003	0.833 $\pm$ 0.003	0.768 $\pm$ 0.001	0.816 $\pm$ 0.003	0.728
	AWQ	6.468 $\pm$ 0.002	11.444 $\pm$ 0.008	0.469 $\pm$ 0.004	0.786 $\pm$ 0.001	0.842 $\pm$ 0.001	0.780 $\pm$ 0.003	0.805 $\pm$ 0.003	0.736
	AWQ + BC	6.467 $\pm$ 0.001	11.450 $\pm$ 0.009	0.459 $\pm$ 0.002	0.787 $\pm$ 0.003	0.843 $\pm$ 0.001	0.787 $\pm$ 0.004	0.815 $\pm$ 0.002	0.738
	AWQ + CLC	6.438 $\pm$ 0.002	11.363 $\pm$ 0.006	0.473 $\pm$ 0.001	0.792 $\pm$ 0.004	0.847 $\pm$ 0.002	0.782 $\pm$ 0.001	0.819 $\pm$ 0.002	0.743
	GPTQ	6.708 $\pm$ 0.004	11.539 $\pm$ 0.005	0.454 $\pm$ 0.002	0.781 $\pm$ 0.002	0.805 $\pm$ 0.002	0.780 $\pm$ 0.002	0.729 $\pm$ 0.001	0.710
	GPTQ + BC	6.709 $\pm$ 0.006	11.538 $\pm$ 0.004	0.451 $\pm$ 0.002	0.779 $\pm$ 0.004	0.805 $\pm$ 0.002	0.778 $\pm$ 0.003	0.729 $\pm$ 0.003	0.708
	GPTQ + CLC	6.702 $\pm$ 0.003	11.534 $\pm$ 0.006	0.459 $\pm$ 0.002	0.779 $\pm$ 0.002	0.814 $\pm$ 0.003	0.779 $\pm$ 0.004	0.787 $\pm$ 0.002	0.724
	FlatQuant	6.520 $\pm$ 0.003	11.394 $\pm$ 0.008	0.476 $\pm$ 0.002	0.793 $\pm$ 0.002	0.838 $\pm$ 0.002	0.780 $\pm$ 0.003	0.801 $\pm$ 0.003	0.738
	FlatQuant + BC	6.516 $\pm$ 0.004	11.394 $\pm$ 0.010	0.482 $\pm$ 0.003	0.793 $\pm$ 0.004	0.839 $\pm$ 0.001	0.782 $\pm$ 0.004	0.801 $\pm$ 0.001	0.739
	FlatQuant + CLC	6.512 $\pm$ 0.005	11.372 $\pm$ 0.007	0.485 $\pm$ 0.002	0.796 $\pm$ 0.003	0.841 $\pm$ 0.004	0.788 $\pm$ 0.002	0.834 $\pm$ 0.001	0.749
LLaMA-3-8B	FP16	5.442 $\pm$ 0.004	8.346 $\pm$ 0.007	0.505 $\pm$ 0.001	0.800 $\pm$ 0.002	0.813 $\pm$ 0.001	0.796 $\pm$ 0.004	0.696 $\pm$ 0.003	0.722
	RTN	6.057 $\pm$ 0.007	9.585 $\pm$ 0.004	0.479 $\pm$ 0.004	0.780 $\pm$ 0.004	0.791 $\pm$ 0.003	0.784 $\pm$ 0.002	0.628 $\pm$ 0.002	0.692
	RTN + BC	6.071 $\pm$ 0.006	9.328 $\pm$ 0.005	0.481 $\pm$ 0.002	0.782 $\pm$ 0.003	0.794 $\pm$ 0.003	0.780 $\pm$ 0.004	0.639 $\pm$ 0.003	0.695
	RTN + CLC	6.024 $\pm$ 0.007	9.252 $\pm$ 0.003	0.490 $\pm$ 0.002	0.781 $\pm$ 0.002	0.796 $\pm$ 0.003	0.782 $\pm$ 0.001	0.668 $\pm$ 0.004	0.703
	AWQ	5.910 $\pm$ 0.007	9.240 $\pm$ 0.007	0.488 $\pm$ 0.001	0.784 $\pm$ 0.003	0.795 $\pm$ 0.002	0.791 $\pm$ 0.002	0.646 $\pm$ 0.002	0.701
	AWQ + BC	5.906 $\pm$ 0.008	9.234 $\pm$ 0.006	0.489 $\pm$ 0.003	0.784 $\pm$ 0.003	0.796 $\pm$ 0.001	0.792 $\pm$ 0.004	0.617 $\pm$ 0.002	0.696
	AWQ + CLC	5.885 $\pm$ 0.007	9.199 $\pm$ 0.005	0.493 $\pm$ 0.004	0.788 $\pm$ 0.001	0.799 $\pm$ 0.002	0.788 $\pm$ 0.002	0.679 $\pm$ 0.002	0.709
	GPTQ	6.011 $\pm$ 0.005	9.370 $\pm$ 0.008	0.487 $\pm$ 0.001	0.783 $\pm$ 0.002	0.798 $\pm$ 0.003	0.788 $\pm$ 0.001	0.689 $\pm$ 0.003	0.709
	GPTQ + BC	6.014 $\pm$ 0.006	9.371 $\pm$ 0.009	0.486 $\pm$ 0.002	0.783 $\pm$ 0.003	0.795 $\pm$ 0.003	0.787 $\pm$ 0.002	0.686 $\pm$ 0.003	0.707
	GPTQ + CLC	6.013 $\pm$ 0.004	9.354 $\pm$ 0.007	0.489 $\pm$ 0.004	0.790 $\pm$ 0.001	0.795 $\pm$ 0.003	0.788 $\pm$ 0.002	0.693 $\pm$ 0.002	0.711
	FlatQuant	6.113 $\pm$ 0.004	9.772 $\pm$ 0.005	0.471 $\pm$ 0.004	0.777 $\pm$ 0.001	0.796 $\pm$ 0.001	0.776 $\pm$ 0.004	0.678 $\pm$ 0.003	0.700
	FlatQuant + BC	6.114 $\pm$ 0.006	9.801 $\pm$ 0.004	0.466 $\pm$ 0.003	0.773 $\pm$ 0.001	0.798 $\pm$ 0.002	0.779 $\pm$ 0.001	0.678 $\pm$ 0.001	0.699
	FlatQuant + CLC	6.105 $\pm$ 0.005	9.747 $\pm$ 0.006	0.473 $\pm$ 0.004	0.782 $\pm$ 0.003	0.794 $\pm$ 0.003	0.788 $\pm$ 0.001	0.686 $\pm$ 0.004	0.705

accuracy from 0.680 to 0.690. On LLaMA-3-8B, AWQ+CLC reduces WikiText-2 PPL from 8.136 to 7.973 and C4 PPL from 13.293 to 12.977, and improves average accuracy from 0.641 to 0.648. This pattern is important because AWQ already uses activation-aware information before the final projection; CLC still provides an additional gain by correcting the remaining lattice-level residual rather than replacing the base quantizer.

For GPTQ and FlatQuant, the absolute perplexity gains are smaller, as expected, because these methods leave less residual error for a post-hoc correction to remove. Nevertheless, CLC still consistently improves the aggregate metrics. On Qwen2.5-7B, GPTQ+CLC improves the average score from 0.674 to 0.678, while FlatQuant+CLC improves it from 0.725 to 0.736. On LLaMA-3-8B, GPTQ+CLC improves the average from 0.633 to 0.636, and FlatQuant+CLC improves it from 0.670 to 0.673. These smaller but consistent gains support the interpretation of CLC as a complementary final-stage correction: when the base quantizer already removes much of the reconstruction error, CLC still recovers the structured component in-

duced by the final integer assignment.

The comparison with classical bias correction further highlights why direct lattice correction is preferable in bias-less LLMs. BC sometimes improves perplexity, but its downstream behavior is less reliable. For example, on LLaMA-3-8B with RTN, BC reduces perplexity but lowers the average downstream score from 0.617 to 0.592, whereas CLC improves it to 0.629. On Qwen2.5-7B with AWQ, BC reduces the average score from 0.680 to 0.662, mainly due to a large RTE drop from 0.678 to 0.614, while CLC improves the average to 0.690. Similarly, on LLaMA-3-8B with FlatQuant, BC improves WikiText-2 PPL but worsens C4 PPL and drops average accuracy from 0.670 to 0.653; CLC improves both perplexity metrics and raises the average to 0.673. These cases show that output-side bias compensation can be decoupled from task robustness, whereas CLC modifies the quantized weights themselves and better aligns perplexity improvement with downstream accuracy.

Overall, the 3-bit additional-model results reinforce the central claim of the paper: CLC is most beneficial when the final projection error is large,

Table 7: 3-bit AdaRound weight-only quantization results. PPL is reported on WikiText-2 and C4 ($\downarrow$); downstream accuracy and average are reported over five lm-eval tasks ($\uparrow$)

Model	Method	PPL($\downarrow$)		Downstream Accuracy($\uparrow$)					
		Wiki	C4	arc-c	arc-e	boolq	piqa	rte	Avg
Mistral-7B	FP16	4.845 $\pm$ 0.001	7.535 $\pm$ 0.002	0.509 $\pm$ 0.002	0.802 $\pm$ 0.001	0.837 $\pm$ 0.003	0.800 $\pm$ 0.003	0.686 $\pm$ 0.004	0.727
	Ada	5.742 $\pm$ 0.006	8.910 $\pm$ 0.005	0.428 $\pm$ 0.003	0.753 $\pm$ 0.002	0.767 $\pm$ 0.002	0.776 $\pm$ 0.003	0.671 $\pm$ 0.002	0.679
	Ada + BC	5.762 $\pm$ 0.005	8.947 $\pm$ 0.005	0.422 $\pm$ 0.004	0.745 $\pm$ 0.002	0.788 $\pm$ 0.003	0.781 $\pm$ 0.003	0.646 $\pm$ 0.003	0.676
	Ada + CLC	5.562 $\pm$ 0.006	8.673 $\pm$ 0.004	0.456 $\pm$ 0.002	0.763 $\pm$ 0.003	0.775 $\pm$ 0.002	0.789 $\pm$ 0.002	0.668 $\pm$ 0.002	0.690
LLaMA-3.1-8B	FP16	5.522 $\pm$ 0.005	8.409 $\pm$ 0.006	0.522 $\pm$ 0.003	0.822 $\pm$ 0.004	0.830 $\pm$ 0.001	0.791 $\pm$ 0.002	0.711 $\pm$ 0.003	0.735
	Ada	10.453 $\pm$ 0.008	17.368 $\pm$ 0.008	0.411 $\pm$ 0.004	0.718 $\pm$ 0.004	0.726 $\pm$ 0.002	0.747 $\pm$ 0.003	0.556 $\pm$ 0.003	0.631
	Ada + BC	10.623 $\pm$ 0.008	17.656 $\pm$ 0.006	0.409 $\pm$ 0.004	0.720 $\pm$ 0.002	0.716 $\pm$ 0.001	0.747 $\pm$ 0.003	0.560 $\pm$ 0.003	0.630
	Ada + CLC	9.810 $\pm$ 0.007	15.870 $\pm$ 0.007	0.415 $\pm$ 0.003	0.733 $\pm$ 0.001	0.746 $\pm$ 0.004	0.751 $\pm$ 0.002	0.585 $\pm$ 0.003	0.646

Table 8: 4-bit AdaRound weight-only quantization results. PPL is reported on WikiText-2 and C4 ($\downarrow$); downstream accuracy and average are reported over five lm-eval tasks ($\uparrow$).

Model	Method	PPL($\downarrow$)		Downstream Accuracy($\uparrow$)					
		Wiki	C4	arc-c	arc-e	boolq	piqa	rte	Avg
Mistral-7B	FP16	4.845 $\pm$ 0.001	7.535 $\pm$ 0.002	0.509 $\pm$ 0.002	0.802 $\pm$ 0.001	0.837 $\pm$ 0.003	0.800 $\pm$ 0.003	0.686 $\pm$ 0.004	0.727
	Ada	5.001 $\pm$ 0.009	7.818 $\pm$ 0.005	0.490 $\pm$ 0.004	0.792 $\pm$ 0.001	0.831 $\pm$ 0.003	0.801 $\pm$ 0.002	0.665 $\pm$ 0.001	0.716
	Ada + BC	5.003 $\pm$ 0.005	7.819 $\pm$ 0.003	0.491 $\pm$ 0.001	0.792 $\pm$ 0.001	0.831 $\pm$ 0.001	0.797 $\pm$ 0.002	0.664 $\pm$ 0.002	0.715
	Ada + CLC	4.992 $\pm$ 0.005	7.806 $\pm$ 0.004	0.493 $\pm$ 0.003	0.792 $\pm$ 0.003	0.834 $\pm$ 0.003	0.800 $\pm$ 0.003	0.664 $\pm$ 0.001	0.717
LLaMA-3.1-8B	FP16	5.522 $\pm$ 0.005	8.409 $\pm$ 0.006	0.522 $\pm$ 0.003	0.822 $\pm$ 0.004	0.830 $\pm$ 0.001	0.791 $\pm$ 0.002	0.711 $\pm$ 0.003	0.735
	Ada	5.980 $\pm$ 0.008	9.627 $\pm$ 0.007	0.497 $\pm$ 0.002	0.790 $\pm$ 0.002	0.785 $\pm$ 0.004	0.786 $\pm$ 0.001	0.621 $\pm$ 0.003	0.696
	Ada + BC	5.984 $\pm$ 0.008	9.631 $\pm$ 0.002	0.501 $\pm$ 0.001	0.792 $\pm$ 0.003	0.789 $\pm$ 0.002	0.787 $\pm$ 0.002	0.625 $\pm$ 0.002	0.699
	Ada + CLC	5.938 $\pm$ 0.002	9.542 $\pm$ 0.002	0.503 $\pm$ 0.004	0.794 $\pm$ 0.002	0.794 $\pm$ 0.002	0.786 $\pm$ 0.004	0.632 $\pm$ 0.002	0.702

but its gains are not limited to RTN or to a single model family. Across Qwen2.5-7B and LLaMA-3-8B, it improves activation-aware, second-order, and transformation-based quantization, showing that budgeted adjacent-level lattice correction is a stable and architecture-preserving way to reduce structured projection residuals.

H.2 Additional Models at 4-bit

Table 6 reports 4-bit results on Qwen2.5-7B and LLaMA-3-8B. This setting is intentionally challenging for demonstrating incremental gains, because 4-bit weight-only PTQ is already relatively strong and the gap to FP16 is smaller than in the low-bit stress setting above. Nevertheless, CLC improves C4 perplexity and average downstream accuracy in every model-quantizer block, and improves WikiText-2 perplexity in seven of the eight blocks. The only exception is GPTQ on LLaMA-3-8B, where WikiText-2 PPL changes marginally from 6.011 to 6.013, while CLC still improves C4 PPL from 9.370 to 9.354 and average accuracy from 0.709 to 0.711. This near-tie is consistent with the main-paper observation that CLC is most visible when the final lattice projection leaves a larger structured residual, but remains beneficial on the more stable C4 and downstream metrics even

for strong 4-bit quantizers.

The results on Qwen2.5-7B show that CLC transfers cleanly beyond the models emphasized in the main paper. Across RTN, AWQ, GPTQ, and FlatQuant, CLC improves both perplexity metrics and the average zero-shot score over the corresponding base quantizer. The gains are particularly clear for downstream accuracy: CLC improves the average score from 0.725 to 0.728 for RTN, from 0.736 to 0.743 for AWQ, from 0.710 to 0.724 for GPTQ, and from 0.738 to 0.749 for FlatQuant. The GPTQ case is especially informative: although the perplexity gain is small, CLC substantially improves RTE from 0.729 to 0.787 and raises the average score by 1.4 points. This suggests that correcting the structured lattice residual can improve downstream behavior even when aggregate language-modeling perplexity appears nearly saturated.

LLaMA-3-8B shows the same qualitative pattern. CLC improves the average zero-shot score for all four quantizers, from 0.692 to 0.703 for RTN, from 0.701 to 0.709 for AWQ, from 0.709 to 0.711 for GPTQ, and from 0.700 to 0.705 for FlatQuant. The RTN and AWQ blocks show the clearest gains: CLC reduces RTN C4 PPL from 9.585 to 9.252 and improves RTE from 0.628 to 0.668, while for

Table 9: Ablation of the CLC knee-point support mask, flip-budget cap, and James-Stein activation-mean estimator on 3-bit AWQ. The *sample mean* variant replaces the James-Stein estimate with the ordinary calibration-set mean. PPL is reported on WikiText-2 and C4 ($\downarrow$); downstream accuracy and average are reported over five LM-Eval-Harness tasks ($\uparrow$). Bold marks the best CLC variant within each model block.

Model	Variant	PPL(↓)		Downstream Accuracy(↑)							Avg
		Wiki	C4	arc-c	arc-e	boolq	piqa	rte			
LLaMA-3.1-8B	AWQ	8.094 ±0.007	13.157 ±0.006	0.413 ±0.002	0.745 ±0.002	0.715 ±0.002	0.756 ±0.001	0.624 ±0.004	0.651		
	w/o knee mask	7.906 ±0.008	12.914 ±0.009	0.408 ±0.002	0.747 ±0.004	0.721 ±0.002	0.762 ±0.002	0.591 ±0.003	0.646		
	w/o budget cap	7.903 ±0.005	12.876 ±0.008	0.409 ±0.003	0.748 ±0.003	0.718 ±0.004	0.763 ±0.003	0.617 ±0.001	0.651		
	w/o JS shrinkage	7.898 ±0.006	12.869 ±0.003	0.414 ±0.002	0.749 ±0.003	0.719 ±0.003	0.763 ±0.003	0.626 ±0.001	0.654		
	CLC (full)	7.893 ±0.007	12.851 ±0.007	0.417 ±0.002	0.749 ±0.001	0.724 ±0.002	0.764 ±0.002	0.638 ±0.003	0.658		
Mistral-7B	AWQ	5.572 ±0.001	8.648 ±0.002	0.439 ±0.003	0.763 ±0.003	0.780 ±0.002	0.787 ±0.001	0.603 ±0.001	0.675		
	w/o knee mask	5.884 ±0.008	8.659 ±0.005	0.432 ±0.001	0.763 ±0.003	0.787 ±0.002	0.783 ±0.002	0.589 ±0.002	0.671		
	w/o budget cap	5.532 ±0.004	8.617 ±0.006	0.438 ±0.002	0.763 ±0.004	0.787 ±0.002	0.785 ±0.004	0.592 ±0.003	0.673		
	w/o JS shrinkage	5.564 ±0.009	8.627 ±0.010	0.439 ±0.004	0.764 ±0.002	0.790 ±0.002	0.785 ±0.002	0.612 ±0.003	0.678		
	CLC (full)	5.524 ±0.005	8.552 ±0.006	0.440 ±0.003	0.766 ±0.004	0.794 ±0.002	0.787 ±0.003	0.617 ±0.001	0.681		

AWQ it reduces C4 PPL from 9.240 to 9.199 and improves the average accuracy to 0.709. These improvements are notable because LLaMA-3-8B is close to the LLaMA-3.1-8B model studied in the main paper, yet CLC continues to provide consistent post-quantization correction.

The comparison with classical bias correction further supports the design choice of correcting the integer weights directly. BC sometimes improves perplexity, but its downstream effect is less reliable. For example, on LLaMA-3-8B with AWQ, BC slightly improves WikiText-2 and C4 PPL but reduces the average score from 0.701 to 0.696, mainly due to a drop on RTE. On Qwen2.5-7B with GPTQ, BC decreases the average score from 0.710 to 0.708, whereas CLC increases it to 0.724. Similarly, for LLaMA-3-8B with FlatQuant, BC worsens both C4 PPL and average accuracy, while CLC improves both. These cases highlight the limitation of output-side compensation: adding or modifying a bias can partially offset a measured mean shift, but it does not change the underlying quantized lattice assignment. CLC instead revises selected integer codes, which better aligns perplexity improvements with downstream robustness.

Overall, the 4-bit additional-model results strengthen the central claim that CLC is complementary to the base quantizer. The gains are not restricted to plain RTN, nor to a single model family: CLC improves activation-aware quantization, second-order quantization, and transformation-based quantization across both Qwen2.5-7B and LLaMA-3-8B. Even in the high-precision 4-bit regime where improvements are expected to be small, CLC consistently improves the most important aggregate metrics, indicating that the struc-

tured first-moment residual addressed by CLC persists across architectures and PTQ pipelines.

H.3 Compatibility with Learned Rounding

Tables 7 and 8 evaluate CLC on top of AdaRound at 3-bit and 4-bit, respectively. This setting tests whether CLC remains useful beyond nearest-level RTN-style projection, since AdaRound already uses calibration data to learn rounding decisions before CLC is applied. The remaining correction space is therefore smaller and more structured: CLC is no longer correcting naive rounding decisions, but revising selected learned lattice choices as a final post-processing step.

The 3-bit results in Table 7 show that CLC is especially helpful when learned rounding is still constrained by a coarse integer lattice. On Mistral-7B, CLC reduces WikiText-2 PPL from 5.742 to 5.562 and C4 PPL from 8.910 to 8.673, while increasing the average downstream score from 0.679 to 0.690. On LLaMA-3.1-8B, the gains are larger: WikiText-2 PPL improves from 10.453 to 9.810, C4 PPL from 17.368 to 15.870, and the average score from 0.631 to 0.646. In contrast, bias correction degrades perplexity and average accuracy in both 3-bit AdaRound settings. This reinforces the same trend observed in the main experiments: directly revising integer codes is a more reliable correction than compensating the output shift with an additional bias term.

At 4-bit, Table 8 shows smaller but consistent gains. On Mistral-7B, CLC reduces WikiText-2 PPL from 5.001 to 4.992 and C4 PPL from 7.818 to 7.806, with the average downstream score increasing from 0.716 to 0.717. On LLaMA-3.1-8B, CLC improves WikiText-2 PPL from 5.980

to 5.938, C4 PPL from 9.627 to 9.542, and average accuracy from 0.696 to 0.702. These improvements are more modest than in the 3-bit setting, as expected, because 4-bit AdaRound already leaves less projection error for a post-hoc correction to remove.

Overall, the AdaRound experiments show that CLC is compatible with learned-rounding quantizers. Although AdaRound changes the interpretation of an adjacent lattice move, the final weights still lie on an integer lattice, allowing CLC to act as a lightweight final-stage correction. The consistent improvements across both bit widths indicate that the structured first-moment residual addressed by CLC persists even after a calibration-driven rounding procedure, especially under more aggressive 3-bit quantization.

H.4 Ablation Study

We ablate the three stabilization components used by CLC: the knee-point support mask, the flip-budget cap, and the James-Stein activation-mean estimator. The knee-point mask determines the reliable support on which lattice corrections are considered, excluding activation-outlier coordinates that are more likely to cause harmful updates. The flip-budget cap controls how aggressively the greedy rule is allowed to apply adjacent-level flips within this support. Finally, the James-Stein estimator replaces the raw calibration-set mean with a shrunk estimate, reducing the effect of noisy coordinate-wise mean estimates when the calibration set is small relative to the activation dimension.

Table 9 reports the ablation on 3-bit AWQ. We compare full CLC against three variants: removing the knee-point mask, removing the flip-budget cap, and replacing the James-Stein mean with the ordinary sample mean. The full CLC configuration gives the best overall trade-off on both models, showing that the three components play complementary roles rather than acting as interchangeable heuristics.

On LLaMA-3.1-8B, full CLC gives the best perplexity and the highest average accuracy among the CLC variants. Removing the knee-point mask or the budget cap still improves perplexity over AWQ, but the average accuracy is weaker than full CLC, mainly due to lower RTE. The sample-mean variant is competitive, yet full CLC further improves the average score from 0.654 to 0.658, while also giving the best WikiText-2 and C4 perplexities. This suggests that James-Stein shrinkage helps stabi-

Table 10: Sensitivity analysis of budget hyperparameter p of CLC on 4-bit Mistral-7B. We sweep p for AWQ and FlatQuant and report WikiText-2/C4 perplexity ($\downarrow$) and PIQA/RTE accuracy ($\uparrow$). Bold marks the best value within each base quantizer.

Base	Setting	Wiki $\downarrow$	C4 $\downarrow$	PIQA $\uparrow$	RTE $\uparrow$
AWQ	Original	4.975 ± 0.003	7.727 ± 0.003	0.797 ± 0.003	0.671 ± 0.004
	$p = 0.01$	4.969 ± 0.006	7.717 ± 0.005	0.798 ± 0.001	0.682 ± 0.003
	$p = 0.02$	4.967 ± 0.003	7.714 ± 0.008	0.798 ± 0.002	0.685 ± 0.001
	$p = 0.03$	4.966 ± 0.005	7.713 ± 0.006	0.801 ± 0.001	0.687 ± 0.001
	$p = 0.04$	4.965 ± 0.003	7.711 ± 0.004	0.802 ± 0.002	0.690 ± 0.004
	$p = 0.05$	4.966 ± 0.007	7.712 ± 0.009	0.799 ± 0.004	0.693 ± 0.002
FlatQuant	Original	5.072 ± 0.005	7.854 ± 0.005	0.794 ± 0.004	0.660 ± 0.002
	$p = 0.01$	5.071 ± 0.003	7.852 ± 0.006	0.794 ± 0.003	0.663 ± 0.003
	$p = 0.02$	5.070 ± 0.005	7.850 ± 0.007	0.795 ± 0.002	0.665 ± 0.001
	$p = 0.03$	5.068 ± 0.004	7.848 ± 0.005	0.796 ± 0.004	0.670 ± 0.003
	$p = 0.04$	5.067 ± 0.008	7.848 ± 0.003	0.799 ± 0.001	0.675 ± 0.002
	$p = 0.05$	5.066 ± 0.005	7.847 ± 0.003	0.797 ± 0.003	0.675 ± 0.002

lize the correction beyond the perplexity objective alone.

On Mistral-7B, the knee-point mask is more critical. Removing it substantially worsens WikiText-2 perplexity relative to both AWQ and full CLC, indicating that flips on activation-outlier coordinates can be harmful. Removing the budget cap and replacing James-Stein shrinkage with the sample mean remain competitive, but both underperform full CLC on perplexity and average accuracy. Overall, the ablation supports the intended roles of the three components: the knee mask filters unsafe coordinates, the budget cap limits over-correction, and James-Stein shrinkage stabilizes the activation mean used for selection.

H.5 Sensitivity to the Flip Budget

We further analyze the sensitivity of CLC to the flip-budget cap p . This budget controls how aggressively CLC applies adjacent-level lattice updates after the knee-point support has been constructed. Importantly, p does not define the correction direction and does not change the candidate support itself; these are determined by the activation-mean statistics and the knee-point rule. Instead, p only limits how many candidate flips the greedy procedure may select within the reliable support. It therefore acts as a safety mechanism: too small a value may leave part of the structured first-moment drift uncorrected, while too large a value may allow the algorithm to consume lower-confidence flips after the dominant bias component has already been reduced.

Table 10 shows that CLC is stable over a small range of flip budgets. For both AWQ and FlatQuant,

increasing p from a very small value improves perplexity, indicating that the first few adjacent-level updates remove a measurable portion of the structured projection error. The gains, however, saturate quickly. For AWQ, the best WikiText-2 and C4 perplexities are obtained around $p = 0.04$, while larger budgets do not provide a clear additional benefit. FlatQuant exhibits the same behavior, with the best or near-best results appearing around $p = 0.04$ - 0.05 .

This saturation is consistent with the design of CLC. The largest first-moment errors are corrected early because the greedy rule prioritizes flips that most directly reduce the estimated channel-wise shift while keeping the post-flip residual controlled. Once these high confidence flips have been selected, the remaining candidates are smaller in effect and less consistently beneficial across evaluation metrics. As a result, allowing more flips does not necessarily translate into better downstream performance. This is visible in the downstream columns: PIQA and RTE improve with moderate budgets, but the improvement does not grow monotonically once the main correction has already been applied.

I Additional Analyses

This appendix collects supplementary analyses that probe CLC beyond the settings of the main paper: transfer to a newer model family on a generation-based reasoning task (App. I.1), robustness to the calibration-set size (App. I.2), the effect of inter-layer error accumulation (App. I.3), and the extension of the formulation to group-wise quantization (App. I.4). The scope of our theoretical guarantees under GPTQ integration is discussed in Appendix G.2.

I.1 Newer Model Family and Generation-based Reasoning

The main-paper evaluation already spans more than two models: in addition to Mistral-7B and LLaMA-3.1-8B, Appendix H reports Qwen2.5-7B and LLaMA-3-8B. To further test whether CLC transfers to a newer model family and to a generation-based reasoning benchmark (rather than multiple-choice zero-shot tasks alone), we evaluate Qwen3-8B on GSM8K using the gsm8k_cot 8-shot generation setting, together with WikiText-2 and C4 perplexity. We focus on weight-only PTQ for LLMs; Qwen3-8B is selected as a suitable re-

Table 11: Qwen3-8B under 4-bit AWQ with and without CLC. Perplexity on WikiText-2 and C4 ($\downarrow$); GSM8K strict-match and flexible-extraction accuracy ($\uparrow$, gsm8k_cot, 8-shot).

Method	Wiki $\downarrow$	C4 $\downarrow$	GSM8K Strict $\uparrow$	GSM8K Flex. $\uparrow$
FP16	8.477	14.174	87.95	88.10
AWQ	8.774	14.624	84.31	83.78
AWQ + CLC	8.773	14.552	85.44	84.46

Table 12: Effect of calibration-set size on LLaMA-3.1-8B with 4-bit AWQ. WikiText-2 and C4 perplexity ($\downarrow$), base quantizer and +CLC.

Calib. size	AWQ Wiki $\downarrow$	+CLC $\downarrow$	AWQ C4 $\downarrow$	+CLC $\downarrow$
FP16	5.522	—	8.409	—
64	5.882	5.862	9.190	9.159
128	5.885	5.862	9.183	9.159
256	5.884	5.862	9.199	9.156
512	5.886	5.863	9.188	9.152

cent LLM architecture for this regime.

Table 11 shows that CLC improves C4 perplexity from 14.624 to 14.552 and, more importantly, improves GSM8K strict-match accuracy from 84.31 to 85.44 and flexible-extraction accuracy from 83.78 to 84.46. These correspond to gains of approximately +1.14 and +0.68 percentage points, recovering roughly one-third of the AWQ-to-FP16 gap in strict-match accuracy. CLC therefore transfers to a newer model family and remains beneficial on a more challenging generation-based reasoning task, and its downstream reasoning gains can exceed what the corresponding perplexity change alone would suggest.

I.2 Sensitivity to the Calibration-set Size

All main-paper experiments use 128 sequences from C4, following the standard calibration setting of GPTQ and AWQ (Frantar et al., 2023; Lin et al., 2024). To assess sensitivity to the calibration budget, we sweep the calibration size over $\{64, 128, 256, 512\}$ sequences on LLaMA-3.1-8B with 4-bit AWQ.

Table 12 shows that CLC improves AWQ at every calibration size, by approximately 0.02 perplexity on WikiText-2 and 0.03 on C4. The gains are nearly unchanged from 64 to 512 sequences, indicating that CLC is not sensitive to the calibration-set size. This stability is consistent with the James–Stein mean estimator (Section E.1), which reduces calibration noise in the activation-mean estimate used for flip selection.

Table 13: Final-layer absolute mean output shift versus FP16 on LLaMA-3.1-8B (4-bit AWQ). E1 runs the quantized network end-to-end (inputs carry upstream errors); E2 feeds FP16 inputs to each quantized block. Lower is better.

Final-layer abs. mean shift	AWQ	AWQ + CLC	CLC reduction
E1 — accumulated	1.095×10^{-3}	6.331×10^{-4}	42.2%
E2 — isolated	4.874×10^{-4}	2.280×10^{-5}	95.3%
Accum. gap: E1 – E2	6.078×10^{-4}	6.103×10^{-4}	—

I.3 Inter-layer Accumulation of the Mean Shift

CLC optimizes an isolated per-layer first-moment error and does not explicitly minimize the accumulated cross-layer output difference. We therefore measure directly whether CLC remains effective when quantization errors propagate through the full network. Using 4-bit AWQ on LLaMA-3.1-8B, we measure the final-layer absolute mean output shift against the FP16 reference under two settings. **E1 (accumulated)** runs the quantized model end-to-end, so each layer receives inputs containing all upstream quantization errors. **E2 (isolated)** feeds the FP16 input to each quantized block, matching the layer-wise setting analyzed by CLC.

Table 13 shows that inter-layer accumulation is non-trivial: for AWQ the accumulated shift is approximately $2.2\times$ the isolated shift. CLC reduces the accumulated end-to-end shift by 42.2%, while nearly eliminating (95.3%) the isolated shift that it directly targets. The remaining accumulation gap is almost unchanged (6.078×10^{-4} vs. 6.103×10^{-4}), indicating that CLC does not explicitly remove cross-layer interactions but also does not amplify them. All perplexity and downstream results in the main paper are evaluated on the fully quantized network and therefore already include these accumulation effects. A theoretically guaranteed propagated-input objective is an important direction for future work.

I.4 Extension to Group-wise Quantization

All reported experiments already use group-wise quantization with group size 128, a standard practical setting in weight-only PTQ. Here we make the group-wise formulation explicit. Each output channel is partitioned into groups, and each weight (i, j) uses the step $s_{g(i,j)}$ of its corresponding group rather than a single per-channel step. CLC applies each adjacent-level flip using this group-dependent step. The first-moment descent argument is un-

changed, because the greedy prefix search still includes the no-flip option. The layer-wise MSE bound extends by replacing the per-channel step with the group-dependent step $s_{g(i,j)}$ throughout, so that the per-flip distortion and the supporting quantities are evaluated with the step of the group each flipped weight belongs to. The current experiments use the standard group size of 128; the same construction applies to other group sizes without modification to the argument.