RCDQ: REFERENCE-GUIDED LLM QUANTIZATION WITH INTERLEAVED CONTINUOUS COMPENSATION AND DISCRETE DECISIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Full-precision reference outputs are widely used to compensate for accumulated errors in low-bit quantization of large language models. Existing methods incorporate these references in different ways. Some apply continuous compensation to an entire layer before quantization. Others incorporate deviations from reference outputs into sequential weight updates or use reference-based reconstruction objectives to optimize quantization parameters and integer codes. A key challenge is that improvements in output reconstruction achieved through continuous compensation may not be preserved after discretization. To address this challenge, we propose **RCDQ** (Reference-Guided Continuous–Discrete Quantization), a two-stage group-wise quantization method that uses the same full-precision reference objective to guide both continuous compensation and quantization decisions. In stage one, before quantizing each weight group, we recompute its weight compensation using statistics describing the output discrepancy between the quantized and full-precision paths. These statistics are updated as groups are processed sequentially. Under the same full-precision reference, in stage two, we then evaluate fake-quantized candidates constructed from the compensated weights and select the best quantization scale multipliers to reduce the discrepancy between the candidate outputs and the full-precision reference outputs. Experiments across seven Llama and Mistral models show that RCDQ achieves the lowest W3 perplexity in all 14 WikiText-2 and C4 model–dataset settings, together with the highest or tied-highest six-task average and MMLU accuracy on six of seven models, respectively. RCDQ remains robust at more aggressive precisions, achieving the lowest W2A16 perplexity in 10 of 14 settings and the lowest W3A8/W4A8 perplexity in three of four settings.

1 INTRODUCTION

Post-training quantization (PTQ) offers a practical way to reduce the memory requirements of large language models (LLMs) using a small calibration set. Layer-wise methods such as GPTQ make low-bit weight quantization feasible at large model scales (Frantar et al., 2023). A central challenge is that quantizing preceding layers changes the inputs received by subsequent layers. Consequently, reconstructing a layer's output on shared calibration inputs does not fully account for its discrepancy from the original full-precision model (Arai & Ichikawa, 2025; Li et al., 2025). Full-precision reference outputs provide a direct target for addressing this accumulated discrepancy.

Existing methods incorporate this reference in several ways. Some derive continuous weight compensation before applying a quantizer, including compensation based on linear-layer and sub-block output matching (Arai & Ichikawa, 2025; Lin & Wan, 2025). Others incorporate discrepancies from the reference into sequential weight updates or alternate error compensation with diffusion to the remaining weights (Li et al., 2025; Zhang et al., 2026; Li et al., 2026). Reconstruction objectives also guide the optimization of quantization parameters and integer codes (Shao et al., 2024; Lee et al., 2026). These approaches establish the importance of output reconstruction and motivate a closer examination of how continuous compensation interacts with discrete quantization decisions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

An improvement in continuous output reconstruction cannot necessarily survive discretization. A compensation update that brings floating-point weights closer to the reference output can alter their dynamic range and their positions relative to representable quantization levels. Subsequent rounding and clipping can therefore weaken the improvement. The choice of quantization scale determines how the compensation is expressed in the discrete weights, making it useful to retain reference information when evaluating those choices. This motivates two connected questions: how can a common full-precision reference guide both continuous compensation and discrete decisions, and which dependencies between these operations must be retained as quantization proceeds?

We present **RCDQ**, **Reference-Guided Continuous–Discrete Quantization**, a two-stage group-wise method that interleaves continuous compensation with reference-guided scale selection. RCDQ maintains an original full-precision path and a current quantization path, and collects statistics describing their output discrepancy. Before quantizing each weight group, stage one applies a local continuous compensation update derived from a regularized reference-reconstruction objective. Stage two then evaluates fake-quantized candidates generated from the compensated weights and selects a scale multiplier for each output channel within the current group. The candidate score combines a curvature-weighted perturbation cost with a linear reference-gradient term. A separate coefficient controls the strength of this linear term, allowing the two stages to use a common reference while controlling how strongly it influences discrete decisions.

RCDQ caches second-order and cross statistics for both stages, avoiding repeated Transformer evaluations for individual candidates. The stage-two score is a local quadratic proxy with the other weight groups fixed. Once the scale multipliers are selected, the resulting per-channel scales are used in a single GPTQ pass for each group, which performs the actual rounding and feedback. The procedure covers all seven linear projections in each decoder block, using linear-output or residual-addition targets according to the projection's position.

We analyze the interaction between the two stages. During quantization, RCDQ uses the same second-order input statistics to choose scale multipliers and to adjust the remaining unquantized weights after each rounding step. As a consequence, rounding one column does not change the reference signal used to compensate later unquantized columns in exact arithmetic. This invariance implies that, in stage one, the entire sequence of continuous compensation updates within a linear layer can be computed in advance, independently of the intervening scale choices and rounding outcomes, while the weights presented to the discrete stage still depend on earlier quantization feedback.

Our contributions are:

- A two-stage group-wise quantization method that carries full-precision reference information through continuous weight compensation and discrete scale multiplier selection.
- An analysis of when the continuous compensation sequence remains independent of interleaved discrete decisions.
- An efficient implementation based on cached second-order and cross statistics, requiring only one actual GPTQ quantization pass per group without repeated Transformer evaluations for individual scale candidates.
- Extensive experiments across seven Llama and Mistral models show that RCDQ achieves the lowest W3 perplexity in all 14 WikiText-2 and C4 model–dataset settings, and the best or tied-best W4 perplexity in 12 of 14 settings. Under W3, it further achieves the highest or tied-highest six-task average and MMLU accuracy on six of seven models, respectively. RCDQ also obtains the lowest W2A16 perplexity in 10 of 14 settings (top-two in 13 of 14) and the lowest perplexity in three of four W3A8/W4A8 settings.

2 Related Work

Efficient post-training quantization. Second-order methods such as OBC and GPTQ establish accurate sequential quantization with curvature-aware rounding and error feedback (Frantar et al., 2022; 2023). Other methods improve quantization through activation-aware scaling or equivalent transformations (Lin et al., 2024; Xiao et al., 2023), and through incoherence processing or rotations (Chee et al., 2023; Ashkboos et al., 2024; Liu et al., 2025; Tseng et al., 2024). RCDQ

instead focuses on the reconstruction objective during group-wise weight quantization, carrying full-precision reference information into both continuous corrections and discrete decisions.

Compensating for accumulated output discrepancies. Quantized preceding layers alter subsequent inputs, motivating reconstruction against full-precision outputs. QEP and LoaQ use such references for continuous output compensation (Arai & Ichikawa, 2025; Lin & Wan, 2025), while GPTAQ, ResComp, and Qronos incorporate propagated discrepancies into sequential quantization updates (Li et al., 2025; 2026; Zhang et al., 2026). RCDQ instead applies a local correction before each group and explicitly evaluates discrete candidates against the same reference, linking continuous compensation to the choice of representable weights.

Reconstruction-guided discrete decisions. Reconstruction objectives have also been used to optimize rounding, blocks, quantizer parameters, and codebooks (Nagel et al., 2020; Li et al., 2021; Shao et al., 2024; Egiazarian et al., 2024). LeanQuant adapts quantization grids using inverse-Hessian sensitivity (Zhang & Shrivastava, 2025), while subsequent methods refine scales or integer codes under reconstruction objectives (Kim et al., 2026; Lee et al., 2026). RCDQ evaluates fake-quantized scale candidates immediately after continuous correction, using a local score that combines a quadratic perturbation cost with a weighted reference-gradient term before actual GPTQ rounding and error feedback.

Structure of continuous correction and error feedback. Our analysis builds on the quadratic structure of second-order quantization (Frantar et al., 2022; 2023) and the triangular interpretation of Qronos (Zhang et al., 2026). With matched curvature, GPTQ feedback preserves the reference gradient on remaining unquantized columns in exact arithmetic. Thus, RCDQ's continuous correction sequence within a linear layer can be precomputed, although the weights used for discrete decisions still depend on earlier rounding outcomes. This separates the analytically determined corrections from the evolving discrete quantization path.

3 METHOD

RCDQ carries a same full-precision reference through two decisions that are often treated separately: a continuous compensation of the weights that are about to be quantized, and a discrete choice among quantization scale multiplier candidates. The two decisions use the same local reference statistics, but have separate coefficients. This section first defines the statistics and local objective, then describes the two stages and their interaction with the actual GPTQ rounding path.

3.1 REFERENCE TRAJECTORIES AND LOCAL OBJECTIVE

We maintain two trajectories through each decoder block. The native trajectory uses the original full-precision model, whereas the current trajectory uses the already quantized preceding blocks and the already quantized projections in the current block. We collect the native outputs before modifying a block and process the projections in a fixed true-sequential order. For a linear projection under consideration, let $W_0 \in \mathbb{R}^{m\times d}$ denote its full-precision weight at the start of its quantization. Let $V \in \mathbb{R}^{m\times d}$ denote the mutable copy of this projection's weight matrix. It is initialized as W_0 and updated as groups are compensated and quantized. Let $X \in \mathbb{R}^{d\times T}$ be the current-trajectory calibration input, and let $Y^\star$ be the corresponding native target. For an ordinary linear output, the target is the output of the same full-precision linear layer. LoaQ (Lin & Wan, 2025) observes that matching only a linear-layer output ignores the residual stream and therefore misses errors accumulated in the full sub-layer output; thus, for o_{proj} and $down_{\mathrm{proj}}$, we form the native target after the corresponding residual addition. In the local decomposition, the current-path residual stream is held fixed while the projection is optimized. Let C denote the tensor added to the current projection output by this residual connection, with $C = 0$ for an ordinary linear target.

The initial output discrepancy between native output and current output is

$$E = Y^\star - (W_0X + C). \tag{1}$$

For a fixed projection input, RCDQ uses the regularized local reconstruction objective

$$J(V) = \|E - (V - W_0)X\|_F^2 + \lambda \|V - W_0\|_F^2 . \tag{2}$$

The gradient of this objective is

$$\nabla_V J(V) = -2\left[EX^\top - (V - W_0)(XX^\top + \lambda I)\right]. \tag{3}$$

For compactness, define the input-induced second-order statistic, the reference cross-statistic, and the negative half-gradient $K(V)$ as

$$A = XX^\top + \lambda I, \qquad B = EX^\top, \qquad K(V) = B - (V - W_0)A. \tag{4}$$

Thus, $\nabla_V J(V) = -2K(V)$. The matrix $XX^\top$ is the input-induced second-order term; λI is the local damping or regularization term. In the default profile, damping is initialized from 0.01 times the mean diagonal of the undamped input statistic, with the same positive-definiteness repairs and Cholesky retries used by the GPTQ executor. All theoretical statements below use the final positive-definite matrix A shared by the continuous and GPTQ updates.

3.2 Stage One: Group-wise continuous compensation

When a group is processed, the current trajectory can already differ from the full-precision reference for two reasons: quantization errors have propagated from earlier projections, and GPTQ rounding errors from earlier groups in the same projection have been fed into the remaining groups. Stage one uses the discrepancy in equation 1 to compensate for these accumulated errors before the current group is discretized. This compensation changes the group center presented to stage two and updates the reference signal used for later groups.

Within a projection, columns are first put in the activation-descending order used by GPTQ (Frantar et al., 2023) and partitioned into groups. Consider a current group G, let V_G denote the submatrix of the current weight matrix V formed by the columns in G, and let R contain all columns that remain free after this group. The current group correction is

$$\Delta_G = \alpha K_G A_{GG}^{-1}, \qquad V_G^+ = V_G + \Delta_G, \tag{5}$$

where α is fixed before quantization ($\alpha = 0.4$ in the default profile). The correction is applied to the current group center; it is not an additional quantization pass and does not modify unprocessed columns directly. The reference statistic needed by later groups is updated as

$$K_{G\cup R}^+ = K_{G\cup R} - \Delta_G A_{G,G\cup R}. \tag{6}$$

The implementation obtains Δ_G by a Cholesky solve rather than an explicit inverse. Only cached A, B, and the current K are needed; computing this correction requires neither a backward pass nor an additional Transformer evaluation.

The continuous step has a simple local descent property. For any $D_G \in \mathbb{R}^{m\times|G|}$, let $[D_G]_G \in \mathbb{R}^{m\times d}$ denote its embedding on the columns in G with zeros elsewhere. Define $V^+ = V + [\Delta_G]_G$. Expanding equation 2 gives

$$J(V + [D_G]_G) - J(V) = \operatorname{tr}\big(D_G A_{GG} D_G^\top\big) - 2\langle D_G, K_G\rangle_F. \tag{7}$$

Substituting $D_G = \Delta_G$ from equation 5 yields

$$J(V^+) - J(V) = -\alpha(2-\alpha)\operatorname{tr}\big(K_G A_{GG}^{-1} K_G^\top\big). \tag{8}$$

Consequently, the continuous compensation does not increase the local objective for $0 \le \alpha \le 2$. Although $\alpha = 1$ gives the exact minimizer of equation 7 when all columns outside the current group are held fixed, this objective is only a local guide: it uses fixed calibration inputs and a projection-level reference rather than the end-to-end model objective. We therefore use the conservative partial update $\alpha = 0.4$ in the default profile, retaining the direction of the local improvement while limiting the size of a compensation that could overfit the calibration reference or over-compensate the weights before discrete quantization. The value 0.4 is a fixed empirical hyperparameter, not a consequence of the local minimization result.

Although the implementation interleaves the two stages, the stage-one compensation sequence within a projection can be computed before the discrete rollouts. The reason is that standard GPTQ feedback based on the same input statistics leaves the reference signal K unchanged on later unquantized columns. Therefore, the stage-one updates depend only on the cached statistics, the group order, and the compensation strength, not on the scale choices or rounding outcomes of preceding groups. Stage two remains sequential: its candidates are formed from weights that include the preceding GPTQ feedback. The formal argument is given in Appendix B.

3.3 Stage Two: Reference-guided decisions in the discrete space

Stage two uses the same native target $Y^\star$ and local statistics as the previous stage. It searches over a fixed candidate set of scale multipliers for weight bit-width b. For each output channel, it applies every candidate to the channel's base scale, constructs the corresponding fake-quantized weights, and evaluates them with the reference-guided local score. It selects the lowest-scoring multiplier for each output channel, then uses the resulting per-channel scales in one actual GPTQ quantization of the current group.

Let $\bar{s}_r$ and z_r be the base scale and zero point for output row r, computed from V_G^+. For an ordered scale multiplier f, RCDQ forms a fake-quantized candidate

$$s_{r,f} = f\bar{s}_r, \qquad Q_{r,G}^{\mathrm{fake}}(f) = s_{r,f}\left[\mathrm{clip}\left(\mathrm{round}\left(\frac{V_{r,G}^+}{s_{r,f}}\right) + z_r, 0, 2^b - 1\right) - z_r\right]. \tag{9}$$

The zero point is kept fixed while the scale is multiplied. Let

$$D_{r,G}(f) = Q_{r,G}^{\mathrm{fake}}(f) - V_{r,G}^+ \tag{10}$$

be the candidate perturbation. Let $K^+ = K(V^+)$ denote the corresponding negative half-gradient evaluated at the compensated weights after stage one. Since this perturbation changes only the current group, the exact change in the same local objective, with all columns outside G fixed, is

$$\Delta J_r(f) = D_{r,G}(f) A_{GG} D_{r,G}(f)^\top - 2\left\langle D_{r,G}(f), K_{r,G}^+ \right\rangle. \tag{11}$$

This equation makes the shared reference explicit: the linear term is computed from K^+, the same reference-gradient statistic produced from $Y^\star$ and used by stage one. RCDQ uses a softened version of this exact increment as its practical score,

$$S_{\beta,r}(f) = D_{r,G}(f) A_{GG} D_{r,G}(f)^\top - 2\beta\left\langle D_{r,G}(f), K_{r,G}^+ \right\rangle, \tag{12}$$

so that

$$S_{\beta,r}(f) = \beta\Delta J_r(f) + (1-\beta) D_{r,G}(f) A_{GG} D_{r,G}(f)^\top. \tag{13}$$

The default profile uses $\beta = 0.05$. A value of $\beta = 1$ would select candidates according to the exact frozen-group increment, but that increment is only a fake-quantization proxy: the selected tensor is subsequently replaced by the actual sequential GPTQ rounding and feedback path. We therefore use a smaller value so that the local objective guides the discrete choice without being treated as the final quantization objective. The coefficient is applied only to the linear reference-gradient term because this term controls how strongly the shared reference favors a particular direction, whereas the quadratic term measures the input-curvature-weighted cost of the candidate perturbation and should remain fully active. Each output row chooses the multiplier with the smallest score. The default profile uses the ordered set of 100 multipliers spanning approximately 0.21 to 1.20. All candidate scores are computed from the cached statistics and the current group, without rerunning the Transformer for each candidate.

3.4 Interleaved execution and actual quantization

After stage one compensates a group and stage two selects its per-channel scale multipliers, RCDQ quantizes the group with one standard GPTQ rollout. RCDQ does not modify the GPTQ rounding or feedback rule; it changes the floating-point weights and scales supplied to that rule through the two stages. After this rollout, the updated weights are used directly as the starting point for the next group.

For compactness, let $\mathcal{Q}_b(W; s, z)$ denote the b-bit fake-quantization operator in Eq. (9), let $\mathrm{QuantParams}(V_{r,G})$ denote the standard per-channel computation of the base quantization scale $\bar{s}_r$ and zero point z_r from the current group weights $V_{r,G}$, and let $\mathrm{GPTQGroup}(\cdot)$ denote one standard GPTQ rollout over the current group using the specified per-channel scales. Algorithm 1 summarizes the complete procedure for one linear projection.

The stage-1-only ablation sets $\alpha = 0.4$ but skips candidate generation and uses the base scale obtained from the compensated center. The stage-2-only ablation sets $\alpha = 0$, while retaining the reference-guided score. The full method uses both stages with $\alpha = 0.4$ and $\beta = 0.05$.

Table 1: Llama model results

Model	Method	#W	Wiki2↓	C4↓	PiQA	Arc E	Arc C	HellaS	WinoG	BoolQ	Avg↑	MMLU↑
Llama-2-7B	FP16	16	5.47	6.97	78.1	75.5	43.2	57.1	69.4	79.4	67.1	45.9
	GPTQ	4	5.63	7.17	77.5	74.7	42.2	56.2	68.5	79.0	66.3	43.6
	GPTAQ	4	5.59	7.14	78.2	75.1	**43.1**	56.1	69.2	78.0	66.6	43.8
	QEP	4	5.63	7.12	78.0	75.2	42.2	56.4	**70.6**	78.6	66.8	44.9
	Qronos	4	5.59	7.15	**78.4**	74.7	42.9	56.1	69.8	78.0	66.7	43.9
	RCDQ	4	**5.56**	**7.10**	78.1	**75.6**	**43.1**	**56.6**	70.0	**79.3**	**67.1**	**45.3**
	FP16	16	5.47	6.97	78.1	75.5	43.2	57.1	69.4	79.4	67.1	45.9
	GPTQ	3	6.41	8.28	73.6	69.1	36.6	49.8	60.9	70.9	60.2	36.8
	GPTAQ	3	6.16	8.06	76.2	70.7	36.4	52.7	65.0	72.0	62.2	37.0
	QEP	3	6.50	7.99	76.5	72.9	**39.7**	52.8	68.1	75.3	64.2	39.0
	Qronos	3	6.00	7.83	76.9	71.9	38.2	53.1	65.9	77.0	63.8	38.3
	RCDQ	3	**5.87**	**7.60**	**77.3**	**73.9**	39.5	**54.4**	**68.4**	**77.5**	**65.2**	**42.9**
Llama-2-13B	FP16	16	4.88	6.47	79.4	78.8	47.5	60.2	72.4	82.5	70.1	55.3
	GPTQ	4	4.98	6.60	78.4	78.5	45.1	59.8	**72.9**	**82.2**	69.5	53.7
	GPTAQ	4	**4.96**	6.58	79.1	78.3	**46.5**	59.7	71.8	81.2	69.4	**54.8**
	QEP	4	4.99	6.57	**79.3**	**78.7**	46.1	**59.9**	72.8	81.4	**69.7**	54.2
	Qronos	4	4.98	6.70	78.9	78.3	46.4	**59.9**	71.2	80.8	69.2	54.3
	RCDQ	4	**4.96**	**6.56**	78.7	78.3	46.2	59.6	72.1	81.7	69.4	53.8
	FP16	16	4.88	6.47	79.4	78.8	47.5	60.2	72.4	82.5	70.1	55.3
	GPTQ	3	5.40	7.17	78.0	76.9	43.2	57.4	**71.1**	80.8	67.9	51.6
	GPTAQ	3	5.33	7.11	78.2	76.9	43.5	57.5	70.3	80.3	67.8	51.5
	QEP	3	5.50	7.07	78.3	76.9	42.6	**57.8**	70.8	81.4	68.0	51.1
	Qronos	3	5.25	7.11	**78.7**	76.9	**44.1**	**57.8**	70.2	80.0	67.9	**52.0**
	RCDQ	3	**5.20**	**6.93**	78.2	**77.4**	43.9	57.7	71.0	**81.8**	**68.3**	51.6
Llama-3-8B	FP16	16	6.14	8.89	78.9	80.8	52.1	60.5	73.4	82.0	71.3	65.5
	GPTQ	4	6.52	9.49	78.2	79.5	48.5	59.3	**74.1**	81.5	70.2	63.4
	GPTAQ	4	6.48	9.45	78.0	78.8	49.7	59.5	74.0	81.1	70.2	62.8
	QEP	4	6.57	9.40	77.6	78.8	49.6	59.4	73.6	**81.6**	70.1	63.5
	Qronos	4	6.45	9.46	**78.9**	80.8	50.5	59.5	71.9	81.2	70.5	**63.6**
	RCDQ	4	**6.39**	**9.29**	78.5	**81.0**	**51.1**	**59.8**	73.5	81.0	**70.8**	**63.6**
	FP16	16	6.14	8.89	78.9	80.8	52.1	60.5	73.4	82.0	71.3	65.5
	GPTQ	3	9.13	12.70	73.6	68.2	37.7	54.2	70.2	73.9	63.0	43.3
	GPTAQ	3	8.39	12.28	73.2	69.2	36.9	55.2	69.4	76.3	63.4	45.6
	QEP	3	8.55	12.21	74.8	70.9	41.5	55.2	68.0	74.8	64.2	46.6
	Qronos	3	7.59	11.70	**77.6**	75.5	41.0	55.6	67.2	76.7	65.6	52.3
	RCDQ	3	**7.17**	**10.83**	**77.6**	**75.8**	**42.3**	**56.9**	**74.0**	**78.2**	**67.5**	**57.7**
Llama-3.1-8B	FP16	16	6.24	8.96	79.3	82.2	52.1	60.7	74.6	83.1	72.0	65.3
	GPTQ	4	6.60	9.56	78.7	79.8	49.9	59.3	73.1	**82.0**	70.5	63.2
	GPTAQ	4	6.56	9.54	78.0	**80.6**	50.6	**59.6**	73.4	81.4	70.6	63.7
	QEP	4	6.65	9.48	78.5	80.1	49.7	59.4	73.3	**82.0**	70.5	**64.2**
	Qronos	4	6.54	9.47	78.7	**80.6**	**50.9**	59.0	73.4	81.5	**70.7**	63.8
	RCDQ	4	**6.48**	**9.38**	**79.0**	**80.6**	50.3	59.5	**73.6**	81.0	**70.7**	64.0
	FP16	16	6.24	8.96	79.3	82.2	52.1	60.7	74.6	83.1	72.0	65.3
	GPTQ	3	8.00	12.15	76.7	75.1	42.2	55.0	70.4	74.6	65.7	51.2
	GPTAQ	3	7.82	11.83	75.4	73.8	40.1	**55.9**	69.5	78.2	65.5	52.1
	QEP	3	8.37	11.87	77.0	74.1	40.9	55.2	70.0	77.1	65.7	50.5
	Qronos	3	7.65	11.22	**78.3**	**75.9**	**42.7**	55.3	71.0	77.9	**66.9**	55.7
	RCDQ	3	**7.24**	**10.90**	78.0	75.1	41.0	**55.9**	**71.4**	**79.8**	**66.9**	**59.6**
Llama-3.2-3B	FP16	16	7.81	10.44	76.3	74.6	43.1	55.8	69.4	73.6	65.5	56.5
	GPTQ	4	8.91	11.20	76.6	73.0	41.6	54.2	69.1	70.2	64.1	52.9
	GPTAQ	4	8.21	11.13	**76.8**	73.3	41.6	54.3	69.0	**74.4**	**64.9**	53.4
	QEP	4	8.67	11.04	76.3	**74.1**	41.8	54.5	68.4	74.1	**64.9**	53.9
	Qronos	4	8.11	11.02	76.1	73.6	40.6	54.1	**69.9**	74.2	64.7	54.0
	RCDQ	4	**8.02**	**10.92**	76.5	73.9	**42.4**	**54.6**	69.0	73.2	**64.9**	**55.6**
	FP16	16	7.81	10.44	76.3	74.6	43.1	55.8	69.4	73.6	65.5	56.5
	GPTQ	3	11.23	14.79	72.4	64.8	35.4	49.3	65.2	70.4	59.6	40.5
	GPTAQ	3	10.03	14.48	72.4	65.5	33.5	49.8	65.7	68.5	59.2	40.5
	QEP	3	11.00	14.05	73.4	67.0	33.5	49.8	64.7	70.5	59.8	42.4
	Qronos	3	10.86	13.36	75.2	64.4	33.4	**50.7**	**66.9**	64.7	59.2	47.2
	RCDQ	3	**8.88**	**12.86**	**75.8**	**69.9**	**37.8**	50.6	66.5	**72.4**	**62.2**	**48.9**

Table 2: Mistral model results

Model	Method	#W	Wiki2↓	C4↓	PiQA	Arc E	Arc C	HellaS	WinoG	BoolQ	Avg↑	MMLU↑
Mistral-7B	FP16	16	5.32	7.84	79.9	80.2	50.8	61.0	74.5	83.5	71.7	62.4
	GPTQ	4	5.44	8.09	**80.4**	79.2	47.2	59.9	72.8	**82.0**	70.2	60.9
	GPTAQ	4	5.42	7.99	80.3	79.0	47.7	**60.1**	**73.4**	81.9	**70.4**	61.0
	QEP	4	5.41	**7.97**	80.1	79.6	48.0	59.1	71.5	80.1	69.7	60.2
	Qronos	4	5.40	7.98	80.1	**79.8**	**48.2**	60.0	73.0	80.6	70.3	**61.3**
	RCDQ	4	**5.39**	7.98	80.0	78.4	46.8	59.9	72.5	81.3	69.8	**61.3**
	FP16	16	5.32	7.84	79.9	80.2	50.8	61.0	74.5	83.5	71.7	62.4
	GPTQ	3	6.31	10.11	77.7	75.1	42.5	57.2	67.8	76.9	66.2	53.0
	GPTAQ	3	6.19	9.47	77.6	74.2	**42.7**	**57.8**	66.9	78.7	66.3	54.7
	QEP	3	6.16	8.93	70.9	60.0	28.6	43.0	61.6	62.2	54.4	32.7
	Qronos	3	5.89	9.10	**78.4**	75.1	**42.7**	57.4	67.1	80.1	66.8	57.0
	RCDQ	3	**5.70**	**8.58**	77.8	**75.2**	41.7	56.5	**71.8**	**81.7**	**67.5**	**57.7**
Mistral-Nemo	FP16	16	5.75	8.43	81.3	83.0	56.8	63.3	75.8	85.8	74.3	69.0
	GPTQ	4	5.96	8.78	80.5	80.7	53.2	62.2	75.8	84.1	72.7	66.7
	GPTAQ	4	5.93	8.75	80.4	82.2	52.7	**62.4**	75.2	**85.7**	**73.1**	**67.4**
	QEP	4	6.03	8.73	80.5	**82.4**	53.4	62.2	**76.2**	83.8	**73.1**	66.7
	Qronos	4	5.95	**8.72**	**80.7**	82.3	**53.8**	61.9	74.7	84.8	**73.1**	67.0
	RCDQ	4	**5.91**	8.73	79.8	82.0	52.6	61.8	76.0	84.3	72.8	66.8
	FP16	16	5.75	8.43	81.3	83.0	56.8	63.3	75.8	85.8	74.3	69.0
	GPTQ	3	7.00	10.43	78.8	76.1	47.2	57.5	68.0	75.2	67.1	59.1
	GPTAQ	3	6.86	10.22	77.4	75.8	46.4	58.6	69.9	79.3	67.9	58.6
	QEP	3	7.33	9.92	78.5	76.4	45.2	**59.0**	**74.0**	80.3	68.9	60.9
	Qronos	3	6.64	9.90	**79.4**	**78.4**	**47.4**	58.8	73.3	**82.2**	**69.9**	61.4
	RCDQ	3	**6.45**	**9.80**	78.3	78.1	46.8	58.0	73.4	81.2	69.3	**61.5**

4 Experiments

4.1 Comparison with Existing Quantization Methods

Tables 1 and 2 compare RCDQ with GPTQ, GPTAQ, QEP, and Qronos across seven language models under W4 and W3 weight-only quantization. Perplexity is evaluated on WikiText-2 (Merity et al., 2016) and C4 (Raffel et al., 2020), with each model calibrated on the corresponding dataset using 128 randomly sampled sequences of length 2,048 and a fixed seed of 0. WikiText-2 samples are contiguous windows from the concatenated training corpus, whereas C4 samples are drawn within individual documents.

For downstream evaluation, we report PIQA (Bisk et al., 2020), ARC-Easy and ARC-Challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2020), BoolQ (Clark et al., 2019), and MMLU (Hendrycks et al., 2021), using models calibrated on C4. Avg. denotes the mean accuracy over the first six tasks. We evaluate Llama-2-7B and Llama-2-13B (Touvron et al., 2023), Llama-3-8B and Llama-3.1-8B (Grattafiori et al., 2024), Llama-3.2-3B (Meta AI, 2024), Mistral-7B (Jiang et al., 2023), and Mistral-Nemo (Mistral AI and NVIDIA, 2024).

Perplexity. Under W3 quantization, RCDQ achieves the lowest perplexity on both WikiText-2 and C4 for all seven models. On Llama-3-8B, for example, it reduces WikiText-2/C4 perplexity from $7.59/11.70$ with the strongest baseline, Qronos, to $7.17/10.83$. Under W4, where quantization error is smaller, RCDQ achieves the best or tied-best perplexity in 12 of 14 model–dataset settings, with only marginal differences on C4 for Mistral-7B and Mistral-Nemo. The larger W3 gains indicate that RCDQ becomes increasingly effective as quantization becomes more aggressive.

Downstream performance. Under W3, RCDQ obtains the highest or tied-highest six-task average and MMLU accuracy on six of the seven models. On Llama-3-8B, it improves the six-task average from 65.6 with Qronos to 67.5, and MMLU from 46.6 with QEP to 57.7. Across the remaining models, the perplexity improvements are generally accompanied by competitive or improved downstream accuracy. Differences are smaller under W4, consistent with the narrower perplexity gaps in this regime.

Table 3: W2A16 perplexity with group size 128. Best and second-best results are shown in **bold** and underline, respectively.

Model	Data	GPTQ	GPTAQ	QEP	Qronos	RCDQ
Llama-2-7B	Wiki	73.37	38.88	47.91	37.12	**13.29**
	C4	96.95	76.45	53.26	**16.98**	18.85
Llama-2-13B	Wiki	21.22	14.36	**12.56**	108.87	16.85
	C4	30.55	23.13	23.05	153.15	**16.72**
Llama-3-8B	Wiki	384.86	149.10	87.29	84.16	**20.95**
	C4	763.96	248.90	121.25	602.40	**29.23**
Llama-3.1-8B	Wiki	210.96	119.37	53.44	26.23	**16.37**
	C4	398.15	188.49	116.27	48.82	**32.98**
Llama-3.2-3B	Wiki	868.48	249.42	96.15	60.15	**22.57**
	C4	505.70	339.34	155.64	130.40	**35.51**
Mistral-7B-v0.3	Wiki	96.97	39.16	25.16	**12.63**	13.87
	C4	91.08	72.74	32.79	**23.88**	25.97
Mistral-Nemo	Wiki	188.81	88.96	28.90	31.98	**14.25**
	C4	281.13	157.10	57.54	41.91	**22.00**

Overall, RCDQ remains competitive under W4 while showing substantially stronger robustness under W3 across various models, language-modeling datasets, and downstream tasks.

4.2 Extreme Low-Bit Quantization

We further evaluate RCDQ under W2A16 quantization with group size 128 across seven models. WikiText-2 and C4 are separately used for calibration and perplexity evaluation. As shown in Table 3, RCDQ achieves the lowest perplexity in 10 of 14 model–dataset settings and ranks among the top two in 13 of 14.

The gains are especially pronounced on the Llama-3 family and Mistral-Nemo, while Qronos or QEP performs best in a few settings. Overall, RCDQ remains the most consistently robust method under the severe discretization error of W2 quantization.

4.3 Extension to Weight–Activation Quantization

We evaluate W3A8 and W4A8 quantization on Llama-2-7B and Llama-3-8B using WikiText-2 perplexity against AWQ (Lin et al., 2024) and QuaRot (Ashkboos et al., 2024) (Table 5).

Table 4: Two-stage ablation on Llama-2-7B (W3).

Configuration	PPL↓
GPTQ Baseline	6.4136
Stage 1 only	6.0690
Stage 2 only	6.0269
Full RCDQ	**5.8689**

Table 5: WikiText-2 perplexity under weight–activation quantization. Lower is better.

Model	Quant.	AWQ Base	QuaRot Base	RCDQ
Llama-2-7B	W3A8	7.296047	6.236552	**6.026924**
Llama-2-7B	W4A8	5.830890	**5.621293**	5.663797
Llama-3-8B	W3A8	10.430760	8.102117	**7.437158**
Llama-3-8B	W4A8	7.026446	6.564392	**6.538615**

RCDQ achieves the lowest perplexity in three of four settings, with the largest gains under W3A8: compared with QuaRot, PPL decreases from 6.2366 to 6.0269 on Llama-2-7B and from 8.1021 to 7.4372 on Llama-3-8B. Under W4A8, RCDQ remains comparable to QuaRot, with differences of at most 0.0425 PPL. These results indicate that RCDQ remains effective with activation quantization, with larger benefits at lower weight precision.

4.4 Ablation of the Two-Stage Design

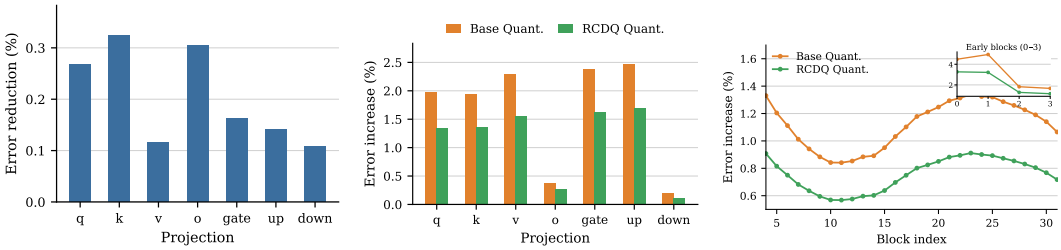
Table 4 ablates the two stages of RCDQ on Llama-2-7B under W3 quantization, using WikiText-2 for both calibration and perplexity evaluation. Stage 1 alone achieves a perplexity of 6.0690, showing that continuous compensation can improve the quantized model even without subsequent scale optimization. Stage 2 alone further reduces the perplexity to 6.0269, indicating that reference-guided discrete scale selection is also effective independently of the continuous correction.

Combining the two stages yields the best perplexity of 5.8689, substantially outperforming either stage used in isolation. This result shows that the two components are complementary rather than redundant: Stage 1 first adjusts the continuous weights toward the full-precision reference, while Stage 2 subsequently selects quantization scales in the discrete space. Their combination allows RCDQ to exploit both continuous compensation and discrete optimization within the same quantization procedure.

4.5 Mechanism Analysis of Continuous Compensation and Discretization

We examine the mechanism underlying RCDQ using Llama-2-7B under W3 quantization with WikiText-2 as the calibration set. Figure 1 separates the effects of continuous compensation and discretization by measuring how Stage 1 changes the reconstruction error and how much error is subsequently introduced by quantization.

Figure 1(a) shows that Stage 1 consistently reduces reconstruction error across all seven projection types, confirming that continuous compensation moves the quantization trajectory toward the full-precision reference. However, Figure 1(b) shows that discretization partially reverses this improvement: using the base quantization scale increases the error for every projection, whereas the reference-guided scale selection in Stage 2 consistently yields a smaller increase. Figure 1(c) further shows that this reduction persists throughout the Transformer depth, including the early blocks shown in the inset. Together, these results directly demonstrate the continuous-to-discrete gap motivating RCDQ: improving the continuous reconstruction alone does not ensure that the gain survives quantization, while Stage 2 preserves more of this improvement after discretization.



(a) Stage one reconstruction error reduction.

(b) Projection-level discretization-induced error.

(c) Block-level discretization-induced error.

Figure 1: **Mechanism analysis of RCDQ.** (a) Reconstruction error reduction from Stage 1. (b) Error introduced by discretization with the base and RCDQ-selected scales. (c) Discretization-induced error across Transformer depth; Blocks 0–3 are shown in the inset and Blocks 4–31 in the main plot.

5 Conclusion

We presented RCDQ, a reference-guided post-training quantization method that couples group-wise continuous compensation with discrete scale selection under a shared full-precision reference. Using cached second-order and cross statistics, RCDQ integrates with standard GPTQ without repeated Transformer evaluations for scale candidates. Experiments across Llama and Mistral models show strong improvements in aggressive low-bit regimes, while ablation and mechanism analyses confirm that the two stages are complementary and that reference-guided scale selection better preserves the gains of continuous compensation after discretization.

AI USAGE DISCLOSURE

In this work, we used generative AI tools for the following purposes: (i) editing and polishing the manuscript to improve grammar, clarity, and readability; (ii) assisting with literature retrieval and the discovery of related work; (iii) supporting research ideation and execution through discussions of the quantization method, algorithm design, and implementation; (iv) drafting portions of the manuscript; and (v) assisting with mathematical derivations and the development of proof arguments. The authors are responsible for reviewing and validating AI-assisted contributions and take full responsibility for the final content of this work, including its text, code, references, mathematical claims, and experimental results.

REFERENCES

Yamato Arai and Yuma Ichikawa. Quantization error propagation: Revisiting layer-wise post-training quantization. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2504.09629.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2404.00456.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. URL https://doi.org/10.1609/aaai.v34i05.6239.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. QuIP: 2-bit quantization of large language models with guarantees. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2307.13304.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2924–2936, 2019. URL https://aclanthology.org/N19-1300/.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12284–12303. PMLR, 2024. URL https://proceedings.mlr.press/v235/egiazarian24a.html.

Elias Frantar, Sidak Pal Singh, and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2208.11580.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.17323.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. URL https://arxiv.org/abs/2310.06825.

Junhan Kim, Gukryeol Lee, Seungwoo Son, Jeewook Kim, and Yongkweon Jeon. Two-stage grid optimization for group-wise quantization of LLMs. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2026. URL https://arxiv.org/abs/2602.02126.

Gunjun Lee, Sehwan Son, Younjoo Lee, Byungjun Kim, and Jung Ho Ahn. SchurQuant: Groupwise discrete optimization for layer-wise LLM quantization. *arXiv preprint arXiv:2608.15567*, 2026. URL https://arxiv.org/abs/2608.15567.

Shuaiting Li, Juncan Deng, Kedong Xu, Rongtao Deng, Hong Gu, Minghan Jiang, Haibin Shen, and Kejie Huang. Rethinking residual errors in compensation-based LLM quantization. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2604.07955.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the limit of post-training quantization by block reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=POWv6hDd9XH.

Yuhang Li, Ruokai Yin, Donghyun Lee, Shiting Xiao, and Priyadarshini Panda. GPTAQ: Efficient finetuning-free quantization for asymmetric calibration. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 36690–36706. PMLR, 2025. URL https://proceedings.mlr.press/v267/li25dn.html.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, 2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/42a452cbafa9dd64e9ba4aa95cc1ef21-Abstract-Conference.html.

Li Lin and Xiaojun Wan. LoaQ: Layer-wise output approximation quantization. *arXiv preprint arXiv:2509.06297*, 2025. URL https://arxiv.org/abs/2509.06297.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM quantization with learned rotations. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2405.16406.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. URL https://arxiv.org/abs/1609.07843.

Meta AI. Llama 3.2. Official model release, 2024. URL https://huggingface.co/meta-llama/Llama-3.2-3B.

Mistral AI and NVIDIA. Mistral nemo. Official model release, 2024. URL https://huggingface.co/mistralai/Mistral-Nemo-Base-2407.

Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *Proceedings of the 37th International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/2004.10568.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL https://jmlr.org/papers/v21/20-074.html.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. URL https://doi.org/10.1609/aaai.v34i05.6399.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. OmniQuant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2308.13137.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. URL https://arxiv.org/abs/2307.09288.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP#: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.04396.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38087–38099. PMLR, 2023. URL https://proceedings.mlr.press/v202/xiao23c.html.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019. URL https://aclanthology.org/P19-1472/.

Shihao Zhang, Haoyu Zhang, Ian Colbert, and Rayan Saab. Qronos: Correcting the past by shaping the future... in post-training quantization. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=7axclBCYul.

Tianyi Zhang and Anshumali Shrivastava. LeanQuant: Accurate and scalable large language model quantization with loss-error-aware grid. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2407.10032.

A RCDQ WORKFLOW FOR ONE LINEAR PROJECTION

Algorithm 1 RCDQ for one linear projection

Require: Calibration input X, native target $Y^\star$, residual term C, initial weight $W_0 \in \mathbb{R}^{m\times d}$, ordered groups $\{G_t\}_{t=1}^{L}$, candidate multipliers $\mathcal{F}$, α, β, bit-width b, and damping λ
Ensure: Quantized weight Q and quantization metadata $\mathcal{M}$

$E \leftarrow Y^\star - (W_0X + C)$
$A \leftarrow XX^\top + \lambda I, \quad B \leftarrow EX^\top$
$V \leftarrow W_0, \quad K \leftarrow B$
$Q \leftarrow 0_{m\times d}, \quad \mathcal{M} \leftarrow \emptyset$
for $t = 1, \dots, L$ **do**
  $G \leftarrow G_t, \quad R \leftarrow \bigcup_{u=t+1}^{L} G_u$
  $\Delta_G \leftarrow \alpha K_G A_{GG}^{-1}$
  $V_G \leftarrow V_G + \Delta_G$
  $K_R \leftarrow K_R - \Delta_G A_{G,R}$
  for $r = 1, \dots, m$ **do**
    $(\bar{s}_r, z_r) \leftarrow \mathrm{QuantParams}(V_{r,G})$
    $D_{r,G}(f) \leftarrow \mathcal{Q}_b\big(V_{r,G}; f\bar{s}_r, z_r\big) - V_{r,G}, \quad \forall f \in \mathcal{F}$
    $f_r^\star \leftarrow \arg\min_{f\in\mathcal{F}} \left[D_{r,G}(f)A_{GG}D_{r,G}(f)^\top - 2\beta\,\langle D_{r,G}(f), K_{r,G}\rangle\right]$
    $s_r^\star \leftarrow f_r^\star \bar{s}_r$
  end for
  $(V, \widehat{Q}_G, \mathcal{M}_G) \leftarrow \mathrm{GPTQGroup}\,(V, G, A, \{s_r^\star, z_r\}_{r=1}^{m})$
  $Q_G \leftarrow \widehat{Q}_G$
  $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{M}_G$
end for
return $Q, \mathcal{M}$

B PRECOMPUTING STAGE-ONE COMPENSATION

The interaction between the two stages admits a useful separation of dependencies. We first state the property used by RCDQ and then explain its scope.

For this analysis, let U be the upper-triangular factor used by standard GPTQ, satisfying

$$A^{-1} = U^\top U. \tag{14}$$

At column j, let v_j be the current column and q_j its rounded value. Standard GPTQ defines $e_j = (v_j - q_j)/U_{jj}$ and applies

$$\delta V = -e_j U_{j,:}, \tag{15}$$

with zeros before column j. This replaces the current column by q_j and feeds its rounding error into the remaining columns.

Lemma 1 (free-column reference-gradient invariance). Assume that A is fixed and positive definite, that the GPTQ feedback uses the factor U in equation 14, and that q_j is any legal rounded value at column j. The matched update equation 15 leaves the reference statistic unchanged on every later column:

$$(\delta K)_{:,k} = 0, \qquad k > j. \tag{16}$$

Proof. Since $K = B - (V - W_0)A$,

$$\delta K = -\delta V A = e_j (UA)_{j,:}.$$

From $A^{-1} = U^\top U$, we have $A = U^{-1}U^{-\top}$, and hence $UA = U^{-\top}$. The matrix $U^{-\top}$ is lower triangular, so its j-th row is zero to the right of the diagonal. This proves equation 16. □

The lemma does not require the rounded value to be globally optimal; it only requires the feedback to match the curvature used in the statistic. It also does not require the scale to be selected independently of the reference. Consequently, reference-guided scale selection cannot invalidate the free-column invariance in exact arithmetic.

Proposition 2 (precomputable continuous correction sequence). Let $G_1, \ldots, G_L$ be a fixed group order, and fix A, B, and α independently of the discrete choices. Under the conditions of Lemma 1, the continuous corrections generated by the interleaved procedure are exactly the solution of the forward recurrence

$$\Delta_{G_t} = \alpha \left(B_{G_t} - \sum_{s<t} \Delta_{G_s} A_{G_s,G_t} \right) A^{-1}_{G_t,G_t}, \qquad t = 1, \ldots, L. \tag{17}$$

In particular, they do not depend on the multipliers, integer codes, or rounding errors selected in preceding groups.

Proof sketch. Initially $K = B$. A continuous update in group G_s subtracts $\Delta_{G_s} A_{G_s,:}$ from the remaining statistic. By Lemma 1, the subsequent GPTQ rollout contributes zero to the free columns of K. Therefore the statistic at G_t is the bracketed expression in equation 17; inserting it into equation 5 gives the recurrence. □

The recurrence in equation 17 means that a complete continuous correction sequence for one projection can be computed before its discrete rollouts. The current implementation evaluates the same recurrence online inside the group loop, which preserves the true-sequential execution and avoids introducing a separate unvalidated executor. The result is a dependency characterization, not a claim that all groups are independent or that the implementation is automatically faster.

The discrete path retains a different dependency. If $F_{<t \to G_t}$ denotes the actual GPTQ feedback from earlier groups, the center seen by the current group has the form

$$V^{+}_{G_t} = W_{0,G_t} + F_{<t \to G_t} + \Delta_{G_t}. \tag{18}$$

Thus the candidate perturbations, base scales, selected multipliers, and final rounded weights remain functions of earlier discrete decisions even though Δ_{G_t} itself is determined by fixed statistics. The separation is local to one projection: the next projection receives a new current input and discrepancy after the preceding quantization, so its own A, B, and correction sequence generally change. The characterization also assumes fixed column order, fixed coefficients, matched curvature, and exact arithmetic; candidate-dependent curvature re-estimation, column reordering, or mismatched feedback falls outside the stated result.

C Fidelity of the Stage-2 Proxy

Stage 2 selects scale multipliers using fake-quantized candidates, whereas the selected scales are ultimately executed through sequential GPTQ rounding and feedback. We therefore examine whether the fake-quant proxy remains reliable under the actual GPTQ rollout. The experiment is conducted on Llama-2-7B under W3A16 quantization with group size 128 using WikiText-2 for calibration. For each row–group decision, we independently execute all 100 candidate multipliers through a diagnostic GPTQ rollout and evaluate the resulting Stage-2 objective. The best candidate among these rollouts is treated as the *grid oracle*. All oracle evaluations are performed on side branches and are discarded afterward, leaving the original RCDQ quantization trajectory unchanged.

Table 6 summarizes the agreement between the multiplier selected by the fake-quant proxy, f_{fake}, and the GPTQ grid oracle, f_{oracle}. Although the exact multiplier matches in only 27.79% of the decisions, the discrepancy is strongly concentrated around the oracle: the median absolute difference is only 0.01, while 89.17% and 96.58% of decisions differ by at most 0.05 and 0.10, respectively.

Table 6: Fidelity of the Stage-2 fake-quant proxy under actual GPTQ rollouts. The oracle is the best candidate among the same 100 multiplier grid points. Lower regret is better.

Metric	Result
Exact multiplier match	27.79%
Mean $\lvert f_{\text{fake}} - f_{\text{oracle}} \rvert$	0.0230
P50 / P90 / P95 / P99 of $\lvert \Delta f \rvert$	0.01 / 0.06 / 0.09 / 0.17
$\lvert \Delta f \rvert \leq 0.05$	89.17%
$\lvert \Delta f \rvert \leq 0.10$	96.58%
Median NRegret	2.05%
P90 / P95 / P99 NRegret	16.82% / 29.05% / 92.56%
NRegret $\leq 10\%$	82.24%
NRegret $\leq 25\%$	93.91%
NRegret $\leq 50\%$	97.61%
Median NRegret (mismatched only)	4.12%
NRegret $\leq 10\%$ (mismatched only)	75.41%

To measure the quality of the multiplier selected by the fake-quant proxy, we compare it against an oracle obtained from actual GPTQ rollouts. For each row–group decision and each candidate multiplier $f \in \mathcal{F}$, let $J_{\text{GPTQ}}(f)$ denote the Stage-2 objective evaluated after applying the corresponding scale and executing the actual GPTQ rounding and feedback for the current group. We denote by f_{fake} the multiplier selected by the fake-quant proxy, and define the grid oracle as

$$f_{\text{oracle}} = \arg\min_{f \in \mathcal{F}} J_{\text{GPTQ}}(f). \tag{19}$$

The candidate $f = 1$ corresponds to the unmodified base quantization scale. We then define the normalized oracle regret as

$$\text{NRegret} = \frac{J_{\text{GPTQ}}(f_{\text{fake}}) - J_{\text{GPTQ}}(f_{\text{oracle}})}{J_{\text{GPTQ}}(1) - J_{\text{GPTQ}}(f_{\text{oracle}}) + \epsilon}, \tag{20}$$

where ϵ is a small constant for numerical stability. Thus, NRegret measures the fraction of the oracle improvement over the base scale that is missed by the fake-quant selection. We use $\epsilon = 10^{-12}$ and exclude decisions whose oracle gain is smaller than 10^{-8} from the normalized statistic. This removes only 48,402 out of 50,593,792 decisions (0.096%), leaving 50,545,390 valid decisions.

The median NRegret is only 2.05%, and 82.24% of valid decisions miss no more than 10% of the improvement available to the grid oracle. Even when restricting the analysis to decisions for which the proxy and oracle select different multipliers, the median NRegret remains 4.12%, with 75.41% of such decisions within 10% of the oracle improvement. These results indicate that sequential GPTQ feedback frequently changes the exact optimal grid point, but the fake-quant proxy generally identifies a nearby candidate whose realized GPTQ objective remains close to the grid oracle.

D Runtime and Memory Overhead

We evaluate the runtime and memory overhead of RCDQ on Llama-2-7B under W3 quantization. As shown in Table 7, RCDQ increases the end-to-end runtime from 646.87 s to 1051.12 s and the core quantization time from 553.18 s to 980.93 s, corresponding to 62.5% and 77.3% overhead, respectively. Peak allocated GPU memory increases from 4.657 GiB to 9.102 GiB, while perplexity decreases from 6.4121 to 5.8689.

Table 7: Runtime, peak memory, and perplexity on Llama-2-7B under W3 quantization.

Method	End-to-end (s)	Core quant. (s)	Peak memory (GiB)	PPL↓
GPTQ	646.87	553.18	4.657	6.4121
RCDQ	1051.12	980.93	9.102	**5.8689**

Table 8 further profiles the Stage-2 scale-selection procedure. Its measured runtime is 84.74 s, approximately 8.64% of the RCDQ core quantization time, with most of the cost coming from reference scoring and candidate selection. The additional peak memory mainly arises from temporary FP64 intermediates used to score the 100 scale candidates; these buffers are required only during offline quantization and do not affect the memory footprint of the final quantized model.

Table 8: Runtime breakdown of Stage-2 scale selection.

Component	**Runtime (s)**
Base scale/zero-point construction	4.82
Generation of 100 candidates	19.98
Reference scoring and selection	59.94
Total	**84.74**