BRACE: ERROR-GUIDED MIXED-PRECISION LLM QUANTIZATION WITHOUT BACKPROPAGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training quantization (PTQ) reduces the deployment costs of large language models (LLMs). Uniform-precision quantization remains a widely used approach, but it often incurs substantial quality degradation at ultra-low bit-widths. In contrast, mixed precision quantization offers a better accuracy–budget trade-off by selectively assigning precision. However, at ultra-low precision, a quantization error cannot be judged by its local magnitude alone. Its significance also depends on the stage-specific response it should induce. This observation motivates a common design principle for bit allocation, error compensation, and quantizer calibration: quantization error should not merely be minimized locally, but should guide the response taken at each stage. Our approach builds on this observation and introduces three key components: (i) Curvature-Conditioned Residual Transport (CCRT) allocates bits using a joint curvature-conditioned cost that combines the error committed at a group with the one-step correction burden transported to its immediate successor; (ii) Block-Anchored Ridge Compensation (BARC) pairs reference and quantized paths within each block to derive a ridge-stabilized correction for propagated error before applying GPTQ to the corrected center under a metric with a ridge term; and (iii) Residual-Feedback Grid Refinement (RFGR) converts the elementwise errors of curvature-weighted first-pass grids into weights for a second scale and zero-point fit at the assigned bit-width, with adjacent-bit error reductions providing the feedback signal. Built on a GPTQ backbone, BRACE uses only a small calibration set and local layer-wise statistics, without backpropagation or iterative retraining. Extensive experiments on language modeling and zero-shot commonsense reasoning across the Llama2, Llama3, Qwen3, and OLMo3 model families show that BRACE achieves state-of-the-art performance among comparable backpropagation-free PTQ methods under 2- and 3-bit weight-only quantization settings. Code will be released upon acceptance.

1 INTRODUCTION

Large language models (LLMs) have massive parameter footprints, imposing storage and data-movement costs during inference, particularly in memory-bound, small-batch settings. Therefore, low-bit weight representations directly reduce these costs, allowing the same model to run within a much smaller hardware budget (Dettmers & Zettlemoyer, 2023; Kim et al., 2023). Post-training quantization (PTQ) offers a practical route to such compression, requiring only a small calibration set and no large-scale retraining. Among PTQ approaches, GPTQ-style second-order reconstruction provides a widely used backbone for weight-only compression (Frantar et al., 2022). Backpropagation-free PTQ is especially attractive at LLM scale because it avoids the repeated forward–backward passes and quantization-time overhead of iterative gradient optimization (Shao et al., 2024; Kim et al., 2026). At ultra-low bit-widths, however, uniform precision leaves an awkward choice: keeping enough bits for the most sensitive groups wastes capacity elsewhere, whereas pushing every group to 2 or 3 bits can sharply degrade model quality. This naturally motivates mixed precision, but its success depends on identifying which groups actually need the extra bits. Moreover, assigning extra precision to one group necessarily changes the budget available to the others, making group-wise bit allocation coupled under the shared budget.

Existing PTQ methods improve low-bit reconstruction through second-order optimization (Frantar & Alistarh, 2022; Frantar et al., 2022), activation-aware scaling, and local calibration (Lin et al.,

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.

2024; Shao et al., 2024). Of these, mixed-precision methods address the allocation side of the problem, typically distributing a fixed budget according to the sensitivity or quantization distortion (Dong et al., 2020; Guan et al., 2024; Li et al., 2023). More recently, LLM quantization has pushed this allocation to the group level, allowing much finer control over how the bit budget is distributed (Huang et al., 2025; Bai et al., 2024; Lee & Song, 2026). Together, these methods provide increasingly refined estimates of local sensitivity and reconstruction error. However, at low precision, these local magnitudes alone do not determine the correction an error demands or the constraints under which that correction can be realized.

For bit allocation, raw local distortion does not show how a candidate residual is conditioned when committed by the sequential quantizer or how much correction it subsequently imposes on the successor. CCRT captures the former through the block-normalized residual $E_g(b)U_{gg}^{-1}$ and the latter through its one-step transport. The same problem remains after the bit map is fixed. Recent work has shown that errors inherited from earlier quantized operations should be incorporated into subsequent reconstruction objectives (Li et al., 2025; Arai & Ichikawa, 2026; Zhang et al., 2026; Zheng et al., 2026). Yet converting the inherited mismatch into a stable correction can be ill-conditioned, and the corrected center must still be discretized under the frozen bit map. At the quantizer level, salience-weighted, output-adaptive, and learned scaling objectives indeed improve local fitting (Huang et al., 2025; Edalati et al., 2025; Wang et al., 2026). Nevertheless, error reductions across adjacent-bit first-pass grids can provide an elementwise feedback signal for refitting the grid at the assigned precision.

We therefore introduce BRACE, a lightweight, backpropagation-free mixed-precision PTQ framework that interprets quantization error through the corrective mechanism available at each stage of allocation, compensation, and quantizer calibration. For allocation, Curvature-Conditioned Residual Transport (CCRT) represents each candidate bit-width by the normalized error committed at the current group and the correction burden transported to its successor. It then minimizes the resulting joint current–future cost under an exact bit budget. Once the bit map is fixed, Block-Anchored Ridge Compensation (BARC) uses paired reference and quantized-path activations to correct errors propagated by earlier operations within the same transformer block. It then quantizes the corrected center with a ridge-aware GPTQ metric. At the quantizer-calibration stage, Residual-Feedback Grid Refinement (RFGR) derives elementwise weights from curvature-weighted error reductions across adjacent-bit first-pass grids and uses them to refit scales and zero-points at the assigned bit-width. Overall, our main contributions are threefold:

- We identify the limitations of characterizing ultra-low-bit quantization error by local magnitude alone and propose BRACE, a lightweight, backpropagation-free framework that interprets error through stage-specific corrective mechanisms for bit allocation, error compensation, and quantizer calibration. We formulate this principle through stage-specific error representations, corrective objectives, and corresponding optimization procedures.
- BRACE comprises three complementary mechanisms. CCRT allocates bits using a cost for each candidate that combines the current-group residual after a group-wise inverse-curvature solve and the correction burden transported to its successor; BARC estimates block-local compensation under the frozen map and quantizes its corrected center using a ridge-adjusted GPTQ curvature; and RFGR performs a second scale and zero-point fit at the assigned precision, weighted by feedback from adjacent-bit error reductions.
- Extensive experiments on Llama2, Llama3, Qwen3, and OLMo3 show that BRACE consistently outperforms the evaluated baselines in perplexity and downstream accuracy under 2- and 3-bit weight-only quantization.

2 RELATED WORKS

2.1 QUANTIZATION OF LLMS

LLM quantization can be organized along two orthogonal axes. One distinguishes whether low-precision constraints are introduced during or after training, and the other whether they are applied to weights alone or to both weights and activations (Xi et al., 2023; Yao et al., 2022; Frantar et al., 2022). Quantization-aware training relies on retraining, whereas post-training quantization (PTQ) operates on a pretrained model using limited calibration data (Hubara et al., 2021; Frantar et al.,

2022). Within PTQ, weight–activation methods reduce both computation and memory traffic but must address activation outliers (Dettmers et al., 2022; Yao et al., 2022; Xiao et al., 2023). BRACE focuses on weight-only PTQ, which reduces model storage and weight-transfer costs without quantizing activations.

Within weight-only PTQ, OBC and GPTQ use approximate second-order information to compensate sequential rounding errors (Frantar & Alistarh, 2022; Frantar et al., 2022). OmniQuant provides a gradient-optimized block-wise calibration baseline, while QuIP# combines incoherence processing with lattice quantization as an alternative low-bit representation (Shao et al., 2024; Tseng et al., 2024). Other approaches improve weight quantizability through activation-aware transformations, sparse high-precision retention, or vector and codebook representations (Lin et al., 2024; Dettmers et al., 2023; Chee et al., 2023; Egiazarian et al., 2024; Liu et al., 2024). These methods improve weight reconstruction, calibration, and low-bit representation in different ways. BRACE builds on GPTQ and focuses on how quantization error should guide the response taken during quantization.

2.2 Mixed-Precision Quantization

Mixed-precision quantization assigns different bit-widths to model components, whereas non-uniform quantization changes the placement of reconstruction levels within a codebook. Prior methods allocate precision at layer, channel, or group granularity using Hessian sensitivity, salience, or quantization distortion (Shen et al., 2020; Dong et al., 2020; Li et al., 2023; Guan et al., 2024). Recent work also combines fine-grained allocation with non-uniform or fractional-bit representations (Zadeh et al., 2020; Bai et al., 2024; Lee & Song, 2026).

SliM-LLM provides the direct group-wise allocation baseline for BRACE. It assigns bit-widths according to group salience and calibrates the corresponding quantizers (Huang et al., 2025). Existing allocation criteria mainly use sensitivity, salience, or local quantization distortion. CCRT instead evaluates each candidate residual after a group-wise inverse-curvature solve and accounts for the correction burden passed to its successor.

2.3 Backpropagation-Free Weight Quantization

Gradient-based PTQ methods, including AdaRound, BRECQ, OmniQuant, and SliderQuant, optimize quantization parameters through iterative updates (Nagel et al., 2020; Li et al., 2021; Shao et al., 2024; Wang et al., 2026). Backpropagation-free PTQ instead relies on local statistics, analytical objectives, and explicit error correction. GPTQ scales Hessian-guided sequential correction to LLMs, and BoA extends this reconstruction to attention blocks (Frantar et al., 2022; Kim et al., 2024). TurboBoA further incorporates propagated-error correction and adaptive grid refinement, making it a direct backpropagation-free baseline for BRACE's compensation and refinement stages (Kim et al., 2026).

Other backpropagation-free methods incorporate quantized-path or accumulated errors into local reconstruction (Li et al., 2025; Arai & Ichikawa, 2026; Zhang et al., 2026; Zheng et al., 2026). This line of work motivates correction beyond isolated layer-wise error, primarily in uniform-precision settings. BRACE instead performs compensation and grid refinement after fixing the mixed-precision map. BARC estimates a ridge-stabilized block-local correction. RFGR uses first-pass errors to reweight the subsequent grid fit at the assigned bit-width.

3 METHOD

BRACE proceeds in two stages. It first uses CCRT to determine and freeze a mixed-precision bit map, and then reloads the pretrained weights to perform BARC and RFGR under that map. The entire process of BRACE is detailed in Figure 1.

3.1 CURVATURE-CONDITIONED RESIDUAL TRANSPORT

Unlike allocation scores based on raw distortion or scalar sensitivity, CCRT first maps each candidate residual through the group-wise triangular solve $E_g(b)U_{gq}^{-1}$. This measures the error committed in the conditional geometry of sequential GPTQ. Transporting the same normalized residual to the

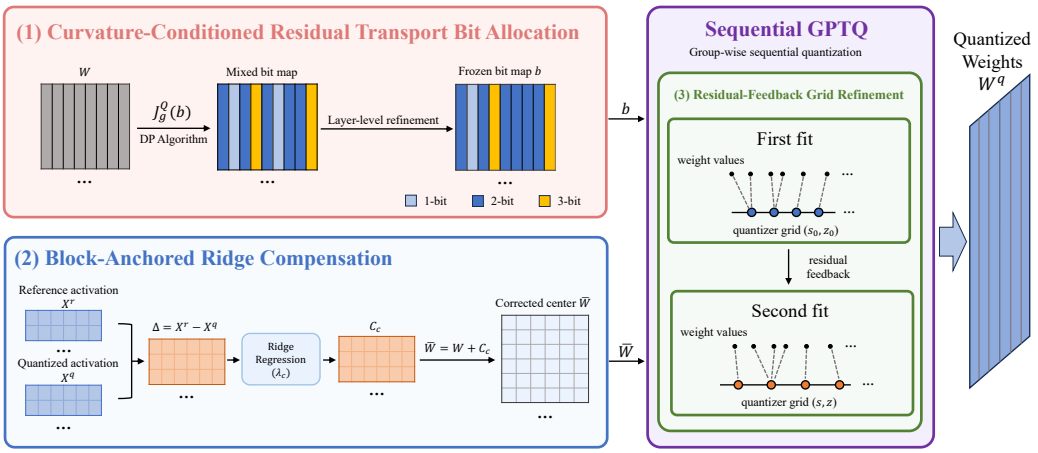


Figure 1: Overview of BRACE. CCRT jointly scores current-group error and transported successor correction to determine the mixed-precision bit map. BARC derives a corrected center from paired reference and quantized-path inputs through ridge regression. Under the frozen map, RFGR uses first-pass error feedback to guide a second grid fit before sequential GPTQ produces the quantized weights.

immediate successor then measures the corresponding correction burden. The resulting joint cost distinguishes bit choices with similar local distortion but different curvature-conditioned current errors or effects on the successor coordinates, and is minimized under the exact bit budget.

For a linear layer with weight matrix $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ and calibration inputs X, define $H = XX^\top + \tau I$ and let U be the upper-triangular factor satisfying $H^{-1} = U^\top U$. The input coordinates are partitioned into groups. For each group, we consider candidate bit-widths $b \in \mathcal{B} = \{\bar{b}-1, \bar{b}, \bar{b}+1\}$, and Q_b denotes the fitted row-wise affine quantizer at bit-width b.

Fix a group g, and let f be its immediate successor. At candidate bit-width b, define $E_g(b) := W_g - Q_b(W_g)$. Scalar GPTQ divides a residual by U_{jj} before transporting it with row j of U. Applying the same triangular solve to a group gives the correction requested from f:

$$C_g(b) := -E_g(b)U_{gg}^{-1}U_{gf}. \tag{1}$$

Here $E_g(b)U_{gg}^{-1}$ is the normalized current residual, and the block right-solve by U_{gg} generalizes division by the scalar GPTQ denominator.

When the successor coordinates are processed, an offset at coordinate j is normalized by U_{jj}. The transported correction is measured in the diagonal conditional metric $D_f := \mathrm{Diag}(U_{jj}^{-2} : j \in f)$. Combining the normalized current residual and the conditionally normalized future correction gives

$$T_g(b) := [-E_g(b)U_{gg}^{-1},\, C_g(b)D_f^{1/2}] = -E_g(b)U_{gg}^{-1}[I_g,\, U_{gf}D_f^{1/2}]. \tag{2}$$

This joint state couples the normalized current residual with its one-step correction demand. The future block is omitted for the final group.

Definition 1 (CCRT transported cost)**.** *For group g at candidate bit-width b, we define*

$$J_g^Q(b) := \|T_g(b)\|_F^2 = \|E_g(b)U_{gg}^{-1}\|_F^2 + \|C_g(b)D_f^{1/2}\|_F^2. \tag{3}$$

The first term evaluates each candidate after the group-wise inverse-curvature solve rather than by raw residual distortion. The second measures the correction burden transported to the successor.

A large $J_g^Q(b)$ indicates a bit choice whose joint state is costly under sequential transport. The CCRT cost admits the following closed-form evaluation:

$$J_g^Q(b) = \mathrm{tr}\big(E_g(b)U_{gg}^{-1}\left(I_g + U_{gf}D_fU_{gf}^\top\right)U_{gg}^{-\top}E_g(b)^\top\big), \tag{4}$$

where the $U_{gf}D_fU_{gf}^\top$ term is omitted for the final group.

The curvature factors are shared across candidate bit-widths, whereas only $E_g(b)$ changes with b. Appendix A.1 gives the derivation.

Evaluating $J_g^Q(b)$ for all group–bit pairs gives the cost table for bit allocation within a Linear:

$$\min_{b_1,\dots,b_G\in\mathcal{B}} \sum_{g=1}^{G} J_g^Q(b_g) \quad \text{s.t.} \quad \sum_{g=1}^{G} b_g = G\bar{b}. \tag{5}$$

We solve Eq. (5) with a multiple-choice dynamic program under the exact average precision $\bar{b}$. This allocation globally minimizes the finite CCRT objective, not downstream task loss. The recurrence, proof, and complexity analysis appear in Appendix A.2.

After solving the per-Linear allocations, we apply a small layer-level refinement. Each Linear's costs are normalized by the calibration output energy $S_\ell = \|W_\ell X_\ell\|_F^2$, and the selected maps are lightly recoupled under a parameter-weighted layer budget. A group either keeps its per-Linear choice $b_{\ell g}^0$ or returns to $\bar{b}$, with at most K changes. Appendix A.3 gives the corresponding program and solver.

3.2 BLOCK-ANCHORED RIDGE COMPENSATION

After CCRT freezes the bit map, BRACE reloads the original weights and applies BARC along the final quantized path. Standard GPTQ quantizes each weight matrix under a reconstruction objective defined by a single set of calibration activations. It does not explicitly pair reference and quantized paths to model the mismatch induced by earlier quantized operations. Within each Transformer block, two paths start from the same block input. The reference path retains full precision weights, whereas the other path is progressively quantized. For a linear layer with weight W, denote their local inputs by X^r and X^q. Their difference reflects the activation discrepancy propagated by earlier quantized operations in the current block.

Let $\Delta = X^r - X^q$. Applying a correction C to the current weight changes the output mismatch to $W\Delta - CX^q$. BARC estimates this correction through ridge regression:

$$\min_C \ \|W\Delta - CX^q\|_F^2 + \lambda_c \|C\|_F^2 . \tag{6}$$

Let $H_q = X^q(X^q)^\top$. For $\lambda_c > 0$, this problem has the unique solution

$$\bar{W} = W + C_c, \qquad C_c = W\Delta(X^q)^\top (H_q + \lambda_c I)^{-1}. \tag{7}$$

The ridge stabilizes the block-local correction C_c by limiting amplification along poorly determined covariance directions. We set $\lambda_c = \gamma\lambda_{\max}(H_q)$, with $\gamma > 0$, to scale the ridge relative to the local covariance. Standard GPTQ damping is added separately for numerical factorization.

We next discretize the corrected center under the frozen map. Completing the square in Eq. (6) yields the quadratic metric $H_q + \lambda_c I$ around $\bar{W}$. Because λ_c stabilizes the correction estimate, BARC uses a separate scale for the ridge term during discretization:

$$H_d = H_q + \lambda_c I. \tag{8}$$

Appendix A.4 derives the center, the induced metric, and the resulting bound on correction amplification.

3.3 RESIDUAL-FEEDBACK GRID REFINEMENT

The frozen map fixes each group's precision. Residual-Feedback Grid Refinement (RFGR) constructs curvature-weighted first-pass grids and obtains their elementwise reconstruction errors. These errors are then converted into weights for refitting the assigned b-bit grid. Thus the second fit depends on the output of the first, without changing the bit map. Positive curvature-weighted error reductions across adjacent bit-widths provide the feedback signal. This signal distinguishes elements with similar current error but different responses to a one-bit change in precision.

Consider a group assigned b bits, and let $\mathcal{N}_b = \{b-1, b+1\} \cap \mathcal{B}$ contain its available adjacent precisions. For $r \in \mathcal{N}_b \cup \{b\}$, let $q_{ij}^{0,r}$ be element (i,j) of the first-pass row-wise affine grid. The superscript 0 marks the first-pass fit, while $q_{ij}^{(b)}(s_i, z_i)$ below denotes the assigned-bit grid as its parameters vary, with $q_{ij}^{0,b} = q_{ij}^{(b)}(s_i^0, z_i^0)$ at the first-pass parameters. Here w_{ij} denotes the current working weight in sequential GPTQ. Each first-pass grid is fitted using the diagonal conditional

weight $h_j = (U^{\mathrm{fin}}_{jj})^{-2}$, where U^{fin} is the final GPTQ factor after incorporating the BARC metric and numerical damping.

$e^{(r)}_{ij} = h_j(w_{ij} - q^{0,r}_{ij})^2$ measures the error of each first-pass grid before the assigned-bit grid is refitted. Let p_{ij} denote the feedback score computed from these errors, as specified below. Denote the group means of p_{ij} and $e^{(b)}_{ij}$ by $\bar{p}$ and $\bar{e}$. For $\bar{p} > 0$, we use $\widehat{p}_{ij} = \min(p_{ij}/\bar{p}, p_{\max})$ and $\beta = \beta_0\bar{p}/(\bar{p} + \bar{e})$.

The assigned-bit refitting objective is

$$\mathcal{L}_{\mathrm{RF}}(s_i, z_i) = \sum_j h_j(1 + \beta\widehat{p}_{ij}) \left(w_{ij} - q^{(b)}_{ij}(s_i, z_i)\right)^2. \tag{9}$$

RFGR refits the scale and zero-point at the assigned bit-width under this objective, keeping the feedback weights fixed during the second fit.

RFGR forms the feedback score from positive error reductions across the first-pass grids:

$$p_{ij} = \frac{1}{|\mathcal{N}_b|} \sum_{r \in \mathcal{N}_b} \left[(b - r)(e^{(r)}_{ij} - e^{(b)}_{ij})\right]_+, \tag{10}$$

where $[x]_+ = \max(x, 0)$. The lower-bit neighbor measures the error avoided at b; the upper-bit neighbor measures the further reduction available at $b + 1$.

Equation (9) is optimized without gradients by alternating nearest-code assignment with an exact weighted least-squares update. We keep the first-pass grid among the candidates and retain a refitted grid only when it lowers $\mathcal{L}_{\mathrm{RF}}$ and satisfies a safeguard on the first-pass reconstruction loss. Quantizer parameterization, solver details, and the fallback property are given in Appendix A.5.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models. We evaluate BRACE on six pretrained base models from four families: Llama2-7B and Llama2-13B (Touvron et al., 2023), Llama3-8B (Grattafiori et al., 2024), Qwen3-8B and Qwen3-14B (Yang et al., 2025), and OLMo-3-7B (Olmo et al., 2025). The evaluated models range from 7B to 14B parameters.

Quantization setup. We evaluate BRACE under weight-only post-training quantization at exact average precisions of 2 and 3 bits. Activations remain unquantized. For the 2- and 3-bit settings, BRACE assigns group precisions from $\{1, 2, 3\}$ and $\{2, 3, 4\}$, respectively, while satisfying the target average bit-width exactly. All methods use a group size of 128 whenever group-wise quantization is supported. BRACE adopts the true-sequential schedule and activation ordering of GPTQ. For calibration, we use 128 sequences of length 2048 randomly sampled from the C4 training set with random seed 0. We set $\gamma = 10^{-3}$, and $K = 2$ for all models and bit-widths. All experiments are conducted on NVIDIA A100 GPUs.

Evaluation. We report perplexity on WikiText2 (Merity et al., 2016) and C4 (Raffel et al., 2020). For commonsense reasoning, we report zero-shot accuracy on PIQA (Bisk et al., 2020), ARC-Easy and ARC-Challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), and BoolQ (Clark et al., 2019). MMLU follows the 5-shot protocol of LM Evaluation Harness v0.4.12 (Biderman et al., 2024). We use length-normalized accuracy for PIQA, ARC-Easy, ARC-Challenge, and HellaSwag, and accuracy for WinoGrande, BoolQ, and MMLU. The reported average is the arithmetic mean of these seven scores.

Baselines. We compare against several representative methods covering different designs in ultra-low-bit weight-only PTQ. Round-to-nearest (RTN) and GPTQ serve as standard rounding and second-order reconstruction baselines (Frantar et al., 2022). We also include OmniQuant for gradient-optimized block-wise calibration and QuIP# for incoherence-processed lattice quantization (Shao et al., 2024; Tseng et al., 2024). SliM-LLM is included as a group-wise mixed-precision method (Huang et al., 2025). TurboBoA incorporates propagated-error correction and adaptive grid

Table 1: Comparison of different quantization methods on WikiText2 and C4. The metric is perplexity, where lower is better. Best results within each quantization setting are shown in bold.

#Bits	Method	Llama2-7B		Llama2-13B		Llama3-8B		Qwen3-8B		Qwen3-14B		OLMo-3-7B	
		Wiki ↓	C4 ↓	Wiki ↓	C4 ↓	Wiki ↓	C4 ↓	Wiki ↓	C4 ↓	Wiki ↓	C4 ↓	Wiki ↓	C4 ↓
FP16	-	5.47	6.97	4.88	6.47	6.14	8.93	7.00	10.37	6.38	9.64	7.92	12.83
INT3	RTN	6.66	8.41	5.52	7.18	12.07	16.49	11.90	16.84	86.10	89.79	9.36	14.41
	GPTQ	6.42	7.91	5.41	7.00	10.64	12.51	8.17	11.49	7.45	10.58	8.94	13.69
	OmniQuant	6.23	7.81	5.41	7.02	8.96	12.67	8.13	11.62	7.15	10.59	8.93	14.00
	QuIP#	6.21	7.70	5.44	6.96	7.78	10.86	7.75	11.23	6.95	10.29	8.45	13.33
	SliM-LLM	6.45	7.85	5.43	7.02	8.10	11.38	8.35	11.73	7.29	10.48	8.77	13.55
	TURBOBOA	6.02	7.48	5.30	6.84	7.39	10.48	7.78	11.22	6.90	10.25	8.42	13.33
	BRACE	**5.93**	**7.42**	**5.22**	**6.80**	**7.31**	**10.33**	**7.64**	**11.00**	**6.79**	**10.11**	**8.37**	**13.28**
INT2	RTN	4.27e3	4.93e3	122.50	139.27	2.76e5	3.04e5	1.10e6	9.93e5	5.62e4	3.39e4	347.76	191.83
	GPTQ	40.54	33.63	28.48	27.44	922.33	216.93	51.00	34.59	25.93	23.49	33.49	28.30
	OmniQuant	16.95	17.31	10.34	11.96	1.26e4	1.29e4	17.87	21.04	13.41	16.94	37.56	37.01
	QuIP#	11.18	12.34	8.59	10.06	20.98	22.39	13.67	17.36	11.01	13.99	12.06	16.59
	SliM-LLM	56.56	18.85	13.08	13.31	257.69	131.05	24.71	24.83	17.67	18.47	19.06	21.43
	TURBOBOA	11.65	12.85	9.26	10.46	18.77	20.92	17.38	19.77	12.62	15.08	12.61	17.03
	BRACE	**8.67**	**9.92**	**7.12**	**8.71**	**15.12**	**16.21**	**11.34**	**14.49**	**9.40**	**12.66**	**11.02**	**15.97**

Table 2: Comparison of different quantization methods on six zero-shot reasoning tasks and 5-shot MMLU. We report length-normalized accuracy (%) for PIQA, ARC-e, ARC-c, and HS, and accuracy (%) for WG, BoolQ, and MMLU. Best results are shown in bold.

Model	#Bits	Method	PIQA ↑	ARC-e ↑	ARC-c ↑	HS ↑	WG ↑	BoolQ ↑	MMLU ↑	Avg ↑
Llama2-13B	FP16	-	79.05	57.91	44.20	76.61	69.61	68.99	55.26	64.52
	INT3	GPTQ	77.91	54.34	40.87	74.09	68.64	71.47	52.08	62.77
		OmniQuant	77.69	55.05	41.89	73.86	68.03	67.43	51.18	62.16
		QuIP#	77.69	56.31	42.24	74.01	68.75	68.47	51.70	62.74
		SliM-LLM	78.07	55.81	42.06	72.96	66.93	65.90	51.23	61.85
		TURBOBOA	78.24	56.48	42.06	73.99	68.73	68.20	52.11	62.83
		BRACE	**78.35**	**58.67**	**43.00**	**74.64**	**69.22**	**71.96**	**52.13**	**64.00**
	INT2	GPTQ	59.36	36.41	27.99	39.57	51.70	57.43	24.23	42.38
		OmniQuant	68.99	42.59	32.59	56.59	54.22	62.32	26.98	49.18
		QuIP#	74.81	49.54	36.52	62.05	65.75	65.44	37.29	55.91
		SliM-LLM	69.31	44.23	32.76	52.41	58.17	**66.97**	28.88	50.39
		TURBOBOA	72.85	46.89	34.22	58.34	65.27	64.68	32.64	53.56
		BRACE	**75.52**	**52.36**	**38.31**	**66.06**	**66.14**	65.63	**40.21**	**57.75**
Qwen3-14B	FP16	-	82.32	81.86	59.22	81.37	74.03	87.06	80.49	78.05
	INT3	GPTQ	79.87	81.90	57.00	78.77	70.72	80.43	73.93	74.66
		OmniQuant	80.03	79.55	53.33	78.17	74.35	84.28	76.41	75.16
		QuIP#	80.63	83.00	57.08	78.34	73.88	84.95	77.37	76.46
		SliM-LLM	79.87	80.09	55.97	79.02	72.61	79.14	76.13	74.69
		TURBOBOA	80.52	**83.25**	57.00	78.50	75.34	86.85	77.43	76.98
		BRACE	**80.71**	81.76	**57.68**	**79.15**	**76.24**	**87.71**	**78.23**	**77.35**
	INT2	GPTQ	60.12	34.09	25.68	47.84	52.88	49.85	25.69	42.31
		OmniQuant	73.78	71.55	42.41	60.61	64.33	63.61	51.24	61.08
		QuIP#	75.37	71.56	46.06	65.30	71.01	76.09	63.00	66.91
		SliM-LLM	70.95	54.97	32.59	60.64	61.96	65.47	42.43	55.57
		TURBOBOA	74.10	67.97	42.24	62.62	67.25	74.10	55.38	63.38
		BRACE	**76.22**	**74.41**	**48.04**	**69.26**	**71.59**	**82.57**	**65.09**	**69.60**

refinement, making it a recent backpropagation-free comparison (Kim et al., 2026). We use its best-performing reported weight-only configuration, which combines TurboBoA with QuaRot (Ashkboos et al., 2024). QuIP# is evaluated without fine-tuning to match the backpropagation-free setting. Further details on baseline reproduction, the common protocol, and model-specific adaptations are provided in Appendix B.

4.2 MAIN RESULTS

Perplexity. Table 1 summarizes the perplexity results on WikiText2 and C4 under 2- and 3-bit weight-only quantization. BRACE achieves the lowest perplexity across all six models, both benchmarks, and both bit-width settings, covering four model families with sizes ranging from 7B to 14B.

Table 3: Ablation study of BRACE under 2-bit weight-only quantization.

Method	Llama2-7B		Llama3-8B		Qwen3-8B	
	Wiki2 (↓)	C4 (↓)	Wiki2 (↓)	C4 (↓)	Wiki2 (↓)	C4 (↓)
w/o CCRT	9.42	11.24	17.00	17.12	13.85	16.13
w/o BARC	10.92	12.64	21.29	22.38	12.72	15.63
w/o BARC ridge	9.40	10.28	16.56	16.96	18.42	19.19
w/o RFGR	13.45	13.58	261.83	189.29	17.30	18.65
w/o feedback refit	9.26	10.41	16.81	16.89	12.56	15.56
BRACE	**8.67**	**9.92**	**15.12**	**16.21**	**11.34**	**14.49**

Table 4: Ablation study of BRACE under 3-bit weight-only quantization.

Method	Llama2-7B		Llama3-8B		Qwen3-8B	
	Wiki2 (↓)	C4 (↓)	Wiki2 (↓)	C4 (↓)	Wiki2 (↓)	C4 (↓)
w/o CCRT	6.00	7.49	7.39	10.40	7.78	11.14
w/o BARC	6.09	7.62	7.83	10.97	7.74	11.12
w/o BARC ridge	5.98	7.48	7.52	10.57	8.37	11.70
w/o RFGR	6.12	7.62	8.20	11.25	7.93	11.25
w/o feedback refit	5.98	7.48	7.39	10.41	7.71	11.09
BRACE	**5.93**	**7.42**	**7.31**	**10.33**	**7.64**	**11.00**

Although the margin varies by model, BRACE consistently improves over the competing methods. Among the baselines, QuIP# is particularly competitive at 2 bits, whereas TurboBoA generally gives the strongest results at 3 bits. The reported weight-only TurboBoA results are obtained with QuaRot (Kim et al., 2026), whereas BRACE does not use an additional rotation transformation. RTN, GPTQ, and OmniQuant show much larger degradation on several models at 2 bits. The advantage of BRACE becomes more pronounced as the precision decreases. For example, on Llama2-7B at 2 bits, BRACE reduces WikiText2 PPL from 11.18 to 8.67 and C4 PPL from 12.34 to 9.92 compared with the best baseline. These correspond to relative reductions of 22.5% and 19.6%, respectively. Similar gains are observed on Llama3-8B, where BRACE lowers perplexity from 18.77 to 15.12 on WikiText2 and from 20.92 to 16.21 on C4. This trend is consistent with our motivation that quantization error should guide the response taken at each stage. At 3 bits, BRACE continues to achieve the lowest perplexity for all models on both benchmarks.

Downstream performance. Table 2 reports the results on six zero-shot commonsense reasoning tasks and 5-shot MMLU. BRACE achieves the highest average accuracy for both Llama2-13B and Qwen3-14B under both 2- and 3-bit quantization. BRACE also gives the best result in 26 of the 28 individual task comparisons. The advantage is particularly pronounced at 2 bits. On Qwen3-14B, BRACE reaches 69.60%, improving the best competing result of 66.91% by 2.69 percentage points. On Llama2-13B, BRACE improves the best competing average from 55.91% to 57.75%. At 3 bits, BRACE retains the highest average accuracy on both models. For example, on Llama2-13B, BRACE achieves 64.00%, compared with 64.52% in FP16.

4.3 ABLATION STUDY

Tables 3 and 4 report ablation results for BRACE under 2- and 3-bit weight-only quantization, respectively. Each variant removes only the specified component or design element while keeping the other components and quantization settings fixed. Across both bit-widths, BRACE has lower WikiText2 and C4 perplexity than every ablated variant on all three models.

Under 2-bit quantization, replacing CCRT with uniform 2-bit allocation under the same average bit budget increases perplexity for every model–dataset pair. On Qwen3-8B, for example, WikiText2/C4 perplexity rises from 11.34/14.49 to 13.85/16.13. The consistent degradation supports CCRT's joint treatment of the error committed at the current group and the correction burden trans-

ported to its successor. Removing BARC has a larger effect than removing CCRT on Llama2-7B and Llama3-8B. On Llama3-8B, perplexity rises from 15.12/16.21 to 21.29/22.38, indicating that activation discrepancies propagated by earlier quantized operations should be accounted for within each Transformer block.

The effect of removing the BARC ridge is model dependent. On Llama2-7B and Llama3-8B, the ridge-free compensation remains better than removing BARC altogether. On Qwen3-8B, however, it raises perplexity from 11.34/14.49 to 18.42/19.19, exceeding the 12.72/15.63 obtained without BARC. This reversal is consistent with over-correction: without ridge stabilization, weakly constrained covariance directions can amplify the propagated mismatch and produce a correction that is difficult to preserve under low-bit discretization. BARC's covariance-scaled ridge limits this amplification and keeps the corrected center compatible with the frozen bit map.

Replacing RFGR with the underlying GPTQ quantizer causes substantial degradation at 2 bits. The Llama3-8B result is a special case and should not be used to quantify the ablation effect of RFGR, since the underlying GPTQ quantizer itself fails in this setting, with WikiText2/C4 perplexity reaching 261.83/189.29. To isolate the contribution of feedback-guided refinement, we retain the curvature-weighted first-pass grid and remove only the second fit. Perplexity increases by 4.2–11.2% in all six comparisons. This controlled comparison identifies an additional contribution from the feedback-guided fit beyond the first-pass grid.

The 3-bit results show the same overall pattern: every leave-one-out variant increases perplexity on all six model–dataset pairs. Removing RFGR has the largest effect on Llama3-8B, where WikiText2/C4 perplexity rises from 7.31/10.33 to 8.20/11.25. Removing the BARC ridge has the largest effect on Qwen3-8B, increasing perplexity from 7.64/11.00 to 8.37/11.70. As at 2 bits, this is worse than removing BARC altogether, which gives 7.74/11.12. The CCRT and feedback-refit ablations also increase each of their six entries by 0.05–0.14 PPL, consistent with the milder overall degradation of 3-bit quantization in Table 1. Overall, CCRT, BARC, and RFGR provide complementary, stage-specific responses to quantization error through bit allocation, propagated-error compensation, and grid refinement, respectively. Appendix C further compares CCRT with SBA and current-error-only allocation, BARC with TurboBoA's F2 compensation, and RFGR with SQC.

4.4 QUANTIZATION EFFICIENCY

Table 5 compares the end-to-end quantization time on a single NVIDIA A100 GPU. The full BRACE pipeline takes 48 minutes for Llama2-7B and 54 minutes for Qwen3-8B. Compared with the other low-bit PTQ baselines, BRACE achieves 3.5×, 1.5×, and 1.9× speedups over SliM-LLM, QuIP#, and TurboBoA, respectively. Although the full pipeline is slower than GPTQ due to the additional bit-allocation stage, about 54% of the total runtime is spent on this one-time CCRT allocation. After the bit map is fixed, the remaining quantization takes about 2.2× the runtime of GPTQ. In addition, inference efficiency after deployment is reported in Appendix D.

Table 5: Quantization time comparison.

Method	Llama2-7B Time (min)	Qwen3-8B Time (min)
GPTQ	10	11
SliM-LLM	167	191
QuIP#	72	85
TurboBoA	91	104
BRACE	48	54
BRACE†	22	25

† Loads a precomputed bit map.

5 CONCLUSION

We introduced BRACE, a backpropagation-free mixed-precision PTQ framework built on the principle that quantization error should not merely be minimized locally, but should guide the response taken at each stage. CCRT allocates bits using a joint curvature-conditioned cost that accounts for both the error committed at the current group and the correction burden transported to its successor, BARC derives ridge-regularized compensation for propagated error, and RFGR feeds errors from the first-pass grids back into quantizer refitting. Extensive experiments on diverse models and benchmarks show that BRACE achieves state-of-the-art performance among comparable backpropagation-free PTQ methods under ultra-low-bit weight-only quantization. Its quantization efficiency and performance facilitate ultra-low-bit LLM deployment.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing of the manuscript and to support literature retrieval and discovery. The selection and use of cited works were determined by the authors. We did not use generative AI tools for research ideation or execution, drafting sections of the paper, generating synthetic datasets, or proving mathematical claims. All AI-assisted writing was reviewed and revised by the authors. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Section 3 specifies the BRACE pipeline and the objectives of CCRT, BARC, and RFGR. Detailed derivations, optimization procedures, and implementation safeguards are provided in Appendices A.1–A.5. The experimental setup in Section 4.1 includes the models, quantization and calibration settings, evaluation protocols, random seed, and hardware used in our experiments. For the reproduced baselines, Appendix B records their provenance and configurations, together with the common comparison protocol and architecture-specific compatibility changes. Our implementation and evaluation scripts will be publicly released upon acceptance.

REFERENCES

Yamato Arai and Yuma Ichikawa. Quantization error propagation: Revisiting layer-wise post-training quantization. *Advances in Neural Information Processing Systems*, 38:151916–151951, 2026.

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

Runsheng Bai, Bo Liu, and Qiang Liu. Skim: Any-bit quantization pushing the limits of post-training quantization. *arXiv preprint arXiv:2412.04180*, 2024.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, et al. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024.

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

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. Quip: 2-bit quantization of large language models with guarantees. *Advances in neural information processing systems*, 36:4396–4429, 2023.

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, Volume 1 (Long and Short Papers)*, pp. 2924–2936, 2019.

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.

Tim Dettmers and Luke Zettlemoyer. The case for 4-bit precision: k-bit inference scaling laws. In *International Conference on Machine Learning*, pp. 7750–7774. PMLR, 2023.

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

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. Spqr: A sparse-quantized representation for near-lossless llm weight compression. *arXiv preprint arXiv:2306.03078*, 2023.

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

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

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

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

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

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

Ziyi Guan, Hantao Huang, Yupeng Su, Hong Huang, Ngai Wong, and Hao Yu. Aptq: Attention-aware post-training mixed-precision quantization for large language models. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, pp. 1–6, 2024.

Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and Xiaojuan Qi. SliM-LLM: Salience-driven mixed-precision quantization for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 25672–25692. PMLR, 13–19 Jul 2025. URL `https://proceedings.mlr.press/v267/huang25aa.html`.

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

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

Junhan Kim, Yeo Jeong Park, Seungwoo Son, Chungman Lee, Ho-young Kim, Joonyoung Kim, and Yongkweon Jeon. Turboboa: Faster and exact attention-aware quantization without backpropagation. *arXiv preprint arXiv:2602.04929*, 2026.

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

Deokjae Lee and Hyun Oh Song. Q-palette: Fractional-bit quantizers toward optimal bit allocation for efficient llm deployment. *Advances in Neural Information Processing Systems*, 38:11525–11558, 2026.

Shiyao Li, Xuefei Ning, Ke Hong, Tengxuan Liu, Luning Wang, Xiuhong Li, Kai Zhong, Guohao Dai, Huazhong Yang, and Yu Wang. Llm-mq: Mixed-precision quantization for efficient llm deployment. In *The Efficient Natural Language and Speech Processing Workshop with NeurIPS*, volume 9, pp. 3, 2023.

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

Yuhang Li, Ruokai Yin, Donghyun Lee, Shiting Xiao, and Priyadarshini Panda. Gptaq: Efficient finetuning-free quantization for asymmetric calibration. *arXiv preprint arXiv:2504.02692*, 2025.

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

Yifei Liu, Jicheng Wen, Yang Wang, Shengyu Ye, Li Lyna Zhang, Ting Cao, Cheng Li, and Mao Yang. Vptq: Extreme low-bit vector post-training quantization for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8181–8196, 2024.

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

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

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

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.

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

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Gao Peng, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 45472–45496, 2024.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Q-bert: Hessian based ultra low precision quantization of bert. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 8815–8821, 2020.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

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

Shigeng Wang, Chao Li, Yangyuxuan Kang, Jiawei Fan, Zhonghong Ou, and Anbang Yao. Sliderquant: Accurate post-training quantization for llms. *arXiv preprint arXiv:2603.25284*, 2026.

Haocheng Xi, Changhao Li, Jianfei Chen, and Jun Zhu. Training transformers with 4-bit integers. *Advances in Neural Information Processing Systems*, 36:49146–49168, 2023.

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

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

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in neural information processing systems*, 35:27168–27183, 2022.

Ali Hadi Zadeh, Isak Edo, Omar Mohamed Awad, and Andreas Moshovos. Gobo: Quantizing attention-based nlp models for low latency and energy efficient inference. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 811–824. IEEE, 2020.

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.

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*, volume 2026, pp. 87775–87798, 2026.

Xingyu Zheng, Haotong Qin, Yuye Li, Haoran Chu, Jiakai Wang, Jinyang Guo, Michele Magno, and Xianglong Liu. First-order error matters: Accurate compensation for quantized large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28883–28891, 2026.

A ADDITIONAL METHOD DERIVATIONS AND PROOFS

A.1 CCRT DERIVATION AND EQUIVALENT QUADRATIC FORM

Let $H = XX^\top + \tau I$ and $H^{-1} = U^\top U$. Since $H = U^{-1}U^{-\top}$, any weight perturbation Δ satisfies

$$\operatorname{tr}(\Delta H \Delta^\top) = \|\Delta U^{-1}\|_F^2. \tag{11}$$

The upper-triangular inverse-curvature factor transports normalized perturbations. In scalar sequential GPTQ, committing $-E_j$ at coordinate j produces the coefficient $-E_j/U_{jj}$. Multiplying it by the remaining entries of row j updates the unprocessed weights. The group-wise counterpart is $-E_g(b)U_{gg}^{-1}$, whose product with U_{gf} yields Eq. (1).

For the future-coordinate weighting, an offset δ_j is normalized by U_{jj} when coordinate j is processed, with conditional quadratic cost $\|\delta_j/U_{jj}\|_2^2$. Aggregating these coordinate-wise costs over f yields

$$\sum_{j\in f}\left\|\frac{\delta_j}{U_{jj}}\right\|_2^2 = \left\|\Delta_f D_f^{1/2}\right\|_F^2, \qquad D_f = \operatorname{Diag}\left(U_{jj}^{-2} : j \in f\right), \tag{12}$$

where Δ_f collects the future residuals. Here D_f is the diagonal conditional metric induced by GPTQ's sequential denominators; it is not a replacement for the full future-block Hessian. It weights $C_g(b)$ in the second block of $T_g(b)$, while the normalized current residual forms the first. Substituting Eq. (2) into Eq. (3) yields

$$\begin{aligned} J_g^Q(b) &= \left\|E_g(b)U_{gg}^{-1}[I_g,\, U_{gf}D_f^{1/2}]\right\|_F^2 \\ &= \operatorname{tr}\left(E_g(b)U_{gg}^{-1}\left(I_g + U_{gf}D_fU_{gf}^\top\right)U_{gg}^{-\top}E_g(b)^\top\right), \end{aligned} \tag{13}$$

which is Eq. (4). For the final group, the future block is absent.

A.2 EXACT FINITE-BUDGET ALLOCATION

Theorem 2 (Exact finite-budget allocation)**.** *The multiple-choice dynamic program returns a global minimizer of Eq. (5) with exact average precision $\bar{b}$. For G groups, it requires $O(G^2\bar{b}\,|\mathcal{B}|)$ time and $O(G^2\bar{b})$ backpointer memory; its value memory can be reduced to $O(G\bar{b})$ using two rows.*

Proof. Let $F_t(s)$ be the minimum CCRT cost attainable by the first t groups using exactly s bits. The implementation uses

$$F_t(s) = \min_{b\in\mathcal{B}:\, b\le s}\left\{F_{t-1}(s-b) + J_t^Q(b)\right\}, \qquad F_0(0) = 0, \tag{14}$$

with infeasible states set to infinity, and backtracks from $F_G(G\bar{b})$. To see exactness, assume inductively that $F_{t-1}(s-b)$ is the cheapest assignment of the first $t-1$ groups with bit sum $s-b$. Every feasible assignment of t groups ends in some $b \in \mathcal{B}$, so minimizing over that last choice enumerates every feasible predecessor and returns the minimum for t groups. The base case follows from $F_0(0) = 0$; induction then shows that the terminal state is a global minimizer of Eq. (5) with exact average precision $\bar{b}$.

There are $O(G\bar{b})$ budget states at each of G stages and $|\mathcal{B}|$ candidate transitions per state, giving $O(G^2\bar{b}|\mathcal{B}|)$ time and $O(G^2\bar{b})$ memory when all backpointers are retained. Two value rows reduce the value memory to $O(G\bar{b})$, proving Theorem 2. □

The implementation additionally subtracts a per-group constant for numerical conditioning and uses distance to a committed-error reference map only to break exact ties; neither operation changes the primary CCRT minimizer.

A.3 CROSS-LINEAR CCRT RECOUPLING

Let $b^0_{\ell g}$ be the independently selected CCRT bit of group g in Linear ℓ, $n_{\ell g}$ its parameter count, and $S_\ell = \|W_\ell X_\ell\|_F^2$ the calibration output energy. The restricted layer-level refinement solves

$$\begin{aligned} \min_{\{b_{\ell g}\}} \quad & \sum_{\ell,g} \frac{J^Q_{\ell g}(b_{\ell g})}{S_\ell} \\ \text{s.t.} \quad & b_{\ell g} \in \{\bar{b}, b^0_{\ell g}\}, \qquad \sum_{\ell,g} n_{\ell g} b_{\ell g} = \bar{b} \sum_{\ell,g} n_{\ell g}, \qquad \sum_{\ell,g} \mathbf{1}[b_{\ell g} \neq b^0_{\ell g}] \leq K. \end{aligned} \tag{15}$$

The independent map is always feasible. We solve the restricted problem with two exact dynamic programs. The first builds the supported rate–distortion envelope for each Linear, and the second combines these envelopes under the parameter-weighted layer budget and global change-count limit. Since each group can choose only its independent bit or the base bit, the refinement cannot introduce an uncertified precision choice. Normalizing by output energy makes costs from differently scaled Linears comparable. The bound K keeps the recoupling local; its default value of 2 permits one balanced pair for equal-mass groups. If the normalized objective does not improve beyond numerical tolerance, the implementation retains the independent map.

Let $\mathcal{A}_{\ell g} = \{\bar{b}, b^0_{\ell g}\}$. The inner state $F_{\ell,t}(s,k)$ is the minimum normalized cost for the first t groups of Linear ℓ. Here s is the parameter-weighted bit sum, and k is the exact number of changes relative to the independent map. With $F_{\ell,0}(0,0) = 0$ and all other initial states infeasible, its recurrence is

$$F_{\ell,t}(s,k) = \min_{b \in \mathcal{A}_{\ell t}} \left\{ F_{\ell,t-1}\big(s - n_{\ell t} b, k - \mathbf{1}[b \neq b^0_{\ell t}]\big) + \frac{J^Q_{\ell t}(b)}{S_\ell} \right\}. \tag{16}$$

The terminal inner states form the envelope $E_\ell(s,k) = F_{\ell,G_\ell}(s,k)$. Let $G_m(s,k)$ be the minimum cost after combining the first m Linears. The outer recurrence is

$$G_m(s,k) = \min_{s',k'} \left\{ G_{m-1}(s-s', k-k') + E_m(s',k') \right\}, \qquad G_0(0,0) = 0, \tag{17}$$

where the minimum ranges over feasible states in the mth envelope. The solver selects the minimum $G_L(B,k)$ over $k \leq K$, with $B = \bar{b} \sum_{\ell,g} n_{\ell g}$. By induction on t, the inner recurrence enumerates every supported allocation within a Linear. Induction on m then shows that the outer recurrence covers every combination of the Linear envelopes. Its selected terminal state is therefore a global minimizer of Eq. (15) over the restricted support. The assignment $b_{\ell g} = b^0_{\ell g}$ for every group is a feasible zero-change state, so falling back to the independent map preserves the exact budget.

A.4 BARC CENTER, DISCRETE METRIC, AND STABILITY BOUND

For the current Linear, the reference-path output and the output obtained by applying a continuous correction C on the quantized path are $Y^r = WX^r$ and $Y^q(C) = (W+C)X^q$, respectively. Hence

$$\begin{aligned} Y^r - Y^q(C) &= WX^r - (W+C)X^q \\ &= W(X^r - X^q) - CX^q = W\Delta - CX^q. \end{aligned} \tag{18}$$

The data-fitting term in Eq. (6) is therefore the squared mismatch between the paired paths. The term $\lambda_c \|C\|_F^2$ is added to regularize the size of the correction, giving the stated ridge objective.

Write $R = \Delta(X^q)^\top$ and recall $H_q = X^q(X^q)^\top$. Differentiating Eq. (6) with respect to C gives

$$C(H_q + \lambda_c I) = WR. \tag{19}$$

Since $H_q + \lambda_c I \succ 0$, right multiplication by its inverse yields C_c in Eq. (7) and proves uniqueness. Let $F(C)$ denote the objective in Eq. (6). Writing $C = C_c + D$ and using the normal equation above gives

$$F(C) = F(C_c) + \operatorname{tr}\big[D(H_q + \lambda_c I)D^\top\big]. \tag{20}$$

For a candidate discrete weight Q, set $C = Q - W$, so that $D = C - C_c = Q - \bar{W}$ with $\bar{W} = W + C_c$. Consequently,

$$F(Q - W) = F(C_c) + \operatorname{tr}\big[(Q - \bar{W})(H_q + \lambda_c I)(Q - \bar{W})^\top\big],$$

which explicitly gives the quadratic metric centered at $\bar{W}$ for discretization. To see the directional effect, let $H_q = V\,\mathrm{Diag}(\mu_k)V^\top$. For every $\mu_k > 0$, ridge multiplies the corresponding unregularized inverse direction by $\mu_k/(\mu_k + \lambda_c)$. These factors generally differ with k, unlike a single scalar applied to the full correction. Directions with $\mu_k = 0$ remain well defined through $\lambda_c > 0$.

After estimating C_c, BARC forms the discrete metric in Eq. (8). In the eigenbasis of H_q, its eigenvalues are $\mu_k + \lambda_c$. The correction norm also satisfies

$$\|C_c\|_F \le \|WR\|_F\,\|(H_q + \lambda_c I)^{-1}\|_2 \le \frac{\|WR\|_F}{\lambda_c}, \tag{21}$$

where the last inequality follows from $H_q \succeq 0$. □

A.5 Residual-feedback grid fitting

For row i and input coordinate j, the affine quantizer used by BRACE is

$$q_{ij}^{(b)}(s_i, z_i) = s_i\left[\mathrm{clip}\big(\mathrm{round}(w_{ij}/s_i + z_i), 0, 2^b - 1\big) - z_i\right]. \tag{22}$$

For fixed integer codes k_{ij}, write $q_{ij} = s_i k_{ij} + c_i$ with $c_i = -s_i z_i$. Given nonnegative weights a_{ij}, the fixed-code subproblem is

$$\min_{s_i, c_i} \sum_j a_{ij}(w_{ij} - s_i k_{ij} - c_i)^2.$$

Setting the derivative with respect to c_i to zero gives $c_i = \bar{w}_i - s_i\bar{k}_i$. Substituting this relation and differentiating with respect to s_i gives the exact weighted least-squares update

$$s_i = \frac{\sum_j a_{ij}(k_{ij} - \bar{k}_i)(w_{ij} - \bar{w}_i)}{\sum_j a_{ij}(k_{ij} - \bar{k}_i)^2}, \qquad c_i = \bar{w}_i - s_i\bar{k}_i, \tag{23}$$

where the bars denote weighted means. BRACE alternates this update with nearest-code reassignment from a finite set of range initializations. The update in Eq. (23) applies when the weighted code variance in its denominator is positive. If that variance is zero, all active codes are identical and the scale is not identifiable from the fixed-code subproblem. The implementation retains the current positive scale and uses $c_i = \bar{w}_i - s_i\bar{k}_i$, which attains the optimal constant reconstruction for those codes. The first-pass grids at $b-1$, b, and $b+1$ take $a_{ij} = h_j$. Refitting candidates use $a_{ij}^{(\rho)} = h_j(1 + \rho\beta\widehat{p}_{ij})$ for $\rho \in \{1, 1/2, 1/4\}$ and are evaluated under the full objective in Eq. (9). We set $\beta_0 = 2$ and $p_{\max} = 10$. When $\bar{p} = 0$, BRACE retains the first-pass grid without refitting.

Let (s_i^0, z_i^0) be the first-pass candidate and let $\mathcal{A}_i$ be the finite candidate set searched for row i. Since $(s_i^0, z_i^0) \in \mathcal{A}_i$ and a new candidate is retained only after a strict decrease,

$$\mathcal{L}_{\mathrm{RF}}(s_i^*, z_i^*) \le \mathcal{L}_{\mathrm{RF}}(s_i^0, z_i^0). \tag{24}$$

Thus residual-feedback refitting cannot increase its fitting objective relative to the first-pass candidate. An accepted candidate must also keep its curvature-weighted reconstruction error within 5% of that of the first-pass grid.

B Baseline reproduction protocol

Result provenance and common protocol. All baseline entries in Tables 1 and 2 come from our runs of the authors' publicly released implementations rather than values copied from the corresponding papers. For each model, BRACE and all baselines use the same pretrained checkpoint and tokenizer. They use the fixed C4 calibration samples drawn with random seed 0, and a common implementation evaluates perplexity and downstream tasks. The resulting calibration and evaluation protocol is identical for every method.

Method configurations. We retain the configurations and recommended hyperparameters provided by the released implementations. We do not tune them separately for individual models or evaluation datasets. The same group size is used whenever a method supports group-wise quantization. We evaluate TurboBoA together with QuaRot because the authors attribute their state-of-the-art weight-only results to the addition of an outlier-suppression method and use this combination in their main comparison (Kim et al., 2026; Ashkboos et al., 2024). QuIP# is run without fine-tuning to keep the comparison backpropagation-free.

Architecture compatibility. Some released implementations do not natively support all models evaluated in our experiments. In such cases, we add only the compatibility code needed for the corresponding architecture. These changes include model and module discovery, grouped-query attention layouts, positional information, and model-specific forward APIs. They do not alter the method's quantization objective or quantizer. The quantization order, bit budget, and recommended hyperparameters also remain unchanged.

Checks under the published protocols. Before using our unified protocol, we verify the baseline implementations under the configurations reported in their respective papers. Our reruns of key methods, including TurboBoA, OmniQuant, and QuIP#, match the published table values within ordinary run-to-run numerical variation (Kim et al., 2026; Shao et al., 2024; Tseng et al., 2024). TurboBoA with QuaRot provides a representative example. For 2-bit Llama3-8B, it obtains WikiText2/C4 perplexities of 13.54/32.99 under the WikiText2-calibration protocol of the TurboBoA paper, matching the reported values. Under the unified C4-calibration protocol used in our main comparison, the same method obtains 18.77/20.92. The main tables therefore use the common-protocol results rather than combining results from separately calibrated models.

C ADDITIONAL ABLATION RESULTS

Table 6: Ablation study of BRACE under 2-bit weight-only quantization.

BRACE variant	Llama2-7B		Llama3-8B		Qwen3-8B	
	Wiki2 (↓)	C4 (↓)	Wiki2 (↓)	C4 (↓)	Wiki2 (↓)	C4 (↓)
BRACE	**8.67**	**9.92**	**15.12**	**16.21**	**11.34**	**14.49**
CCRT → SBA allocation	12.23	12.36	26.25	23.73	13.77	16.91
RFGR → SQC quantizer	10.95	11.64	85.38	54.53	12.72	15.62
BARC → TurboBoA F2	10.35	11.81	19.87	20.54	14.44	15.92
CCRT w/ current error only	8.93	10.38	15.84	16.57	12.01	14.96

As discussed in Section 2, the SBA and SQC mechanisms of SliM-LLM (Huang et al., 2025) are the closest prior counterparts to CCRT and RFGR, respectively. Table 6 replaces each of these BRACE components with its SliM-LLM counterpart while leaving the rest of the pipeline unchanged. With SBA allocation, perplexity is higher than BRACE for every model–dataset pair. SBA also underperforms the uniform allocation in the w/o CCRT variant of Table 3 in five cases; the sole exception is Qwen3-8B on WikiText2 (13.77 versus 13.85). This pattern is consistent with SBA not accounting for how a current-group residual changes the correction required by its successor during sequential quantization. Substituting SQC for RFGR likewise raises all six entries. SQC remains reasonably competitive on Llama2-7B and Qwen3-8B, but its large degradation on Llama3-8B indicates that its salience-weighted calibration is less robust here than RFGR's first-pass-error-guided second fit.

We also replace BARC with TurboBoA's F2 mechanism, the error compensation for pre-quantized layers described in Section 3.2 (Kim et al., 2026). It is a suitable counterpart because both methods target input mismatch propagated from preceding quantized operations with analytical updates that require no backpropagation, while deriving their corrections differently. We retain the CCRT map and fit the RFGR grids before applying this update. In place of BARC's ridge-regression center, ridge-aware metric, and subsequent GPTQ recursion, the variant uses TurboBoA's input-deviation term and attention-aware update of the unquantized output rows without importing its Section 3.3 grid selection and refinement. The results show that all six perplexities exceed those of BRACE. On Llama2-7B, the substitution partially improves the w/o BARC result from 10.92/12.64 to 10.35/11.81, compared with 8.67/9.92 for BRACE. Llama3-8B shows the same partial recovery, from 21.29/22.38 to 19.87/20.54, versus 15.12/16.21 for BRACE. On Qwen3-8B, it is worse than removing BARC, giving 14.44/15.92 rather than 12.72/15.63. Within the fixed BRACE pipeline, this component substitution shows that the alternative rule does not reproduce BARC's gain; it is not a comparison with the complete TurboBoA pipeline.

Allocation based only on the current-group error improves on uniform allocation throughout the table, but does not reach full CCRT in any case. Because this variant removes only the successor term, the comparison isolates the benefit of the transported correction burden beyond the current-group error.

D INFERENCE EFFICIENCY

Table 7: Inference efficiency and perplexity on Llama2-7B under 2-bit weight-only quantization.

Method	WM (GiB)	Token/s (↑)	TTFT (ms) (↓)	TPOT (ms) (↓)	Wiki PPL (↓)	C4 PPL (↓)
GPTQ	2.1	198.62	67.89	4.49	40.54	33.63
BRACE	2.2	172.38	68.83	5.27	8.67	9.92

We evaluate the deployment of Llama2-7B quantized by GPTQ and BRACE at an average of 2 bits on a single NVIDIA RTX 4090, using vLLM v0.8.5 with tensor parallelism set to one. We issue 100 requests sequentially without gaps, each with 512 input tokens and 128 output tokens. GPTQ uses the AutoGPTQ legacy packing format and vLLM's corresponding GPTQ kernel. BRACE uses a mixed-precision packing layout and a custom CUDA kernel integrated into vLLM.

Table 7 shows that BRACE achieves 172.38 token/s versus GPTQ's 198.62 token/s, a 13.21% throughput reduction. Median TTFT remains close to GPTQ (68.83 versus 67.89 ms), while median TPOT increases from 4.49 to 5.27 ms. BRACE requires approximately 0.1 GiB more weight memory, primarily due to per-output group-offset metadata in the current implementation rather than the mixed-bit weight payload itself. BRACE reduces WikiText2 perplexity from 40.54 to 8.67 (78.61%) and C4 perplexity from 33.63 to 9.92 (70.50%). These results demonstrate practical deployment through a standard vLLM serving interface, with substantially better 2-bit model quality at a moderate memory and inference-speed cost.