BEYOND LOCAL RECONSTRUCTION: GEOMETRY-PRESERVING OUTPUT ALIGNMENT FOR BINARY LLM QUANTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in post-training quantization for large language models (LLMs) have demonstrated that low-bit quantization methods can maintain most of the original model performance. Despite such successes, 1-bit quantization remains particularly challenging. A common strategy in 1-bit quantization is to determine binary weights by matching full-precision parameters, following a weight-driven criterion. However, this objective is not directly aligned with the quantized model's objective, which is to preserve the model's output behavior under the impact of quantization. A natural alternative is to adopt output-driven criteria that minimize discrepancies in model outputs using calibration data. Surprisingly, naive output-driven approaches often perform even worse in the 1-bit regime. In this paper, we show that this failure arises from two fundamental issues: error accumulation across layers and, more critically, *anisotropic distortion* of the representation space. Based on this observation, we introduce a geometry-preserving output-alignment framework for 1-bit LLM quantization. First, we formulate output alignment as a *constrained optimization* problem that improves reconstruction while explicitly controlling changes in token similarity, preventing reconstruction gains from being achieved at the expense of representation geometry. Second, we propose a *geometry-aware salient-mask selection* strategy that identifies quantization configurations according to their ability to preserve token relations, rather than relying solely on reconstruction error. These components enable effective output-driven optimization while retaining the structural information lost by naive alignment. Extensive experiments across multiple LLM families and downstream tasks demonstrate consistent improvements over existing 1-bit PTQ methods.

1 INTRODUCTION

Large language models (LLMs) (Wei et al., 2022; Radford et al., 2019; Zhang et al., 2022; Brown et al., 2020) have become a focal point of both academic research and industrial development, thanks to their strong capabilities across a wide range of natural language processing tasks (Hendrycks et al., 2020; Bisk et al., 2020), including question answering (Devlin et al., 2019), machine translation (Fan et al., 2020; Lepikhin et al., 2020), summarization (Zhang et al., 2019; Lewis et al., 2019) and language generation (Radford et al., 2019; Brown et al., 2020). Despite these advances, the massive scale of modern LLMs, often involving billions of parameters, poses substantial challenges for efficient inference and deployment. To address this, the community has explored various compression approaches, such as knowledge distillation (Hinton et al., 2015), network quantization (Choi et al., 2018; Frantar et al., 2023), and pruning (Han et al., 2015). However, many of these approaches depend on large-scale training data and costly retraining, which limits their practicality. In contrast, post-training quantization (PTQ) (Ding et al., 2023; Sun et al., 2025) requires only a small calibration set and modest computational resources, making it a practical choice for compressing LLMs. Despite strong empirical performance at near 4-bit quantization, the most extreme case, 1-bit quantization, remains highly challenging.

Existing 1-bit PTQ methods typically use a layer-wise framework with two stages. First, in the quantization stage, a block of weights is binarized by solving a closed-form objective for the quantized variables, while the remaining weights in the layer are kept in full precision. Second, in the error compensation stage, the residual quantization error is mitigated by updating these remaining full-precision weights. Prior approaches differ mainly in the objective used during the quantization

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.

stage: (1) *weight-driven methods*, which minimize $\|W - \widehat{W}\|$ (Xu et al., 2018; Shang et al., 2023), and (2) *output-driven methods*, which align the outputs of the quantized model with those of the full-precision model by minimizing $\|\widehat{X}W - \widehat{X}\widehat{W}\|$ (Li et al., 2025b), where W, $\widehat{W}$, and $\widehat{X}$ denote the full-precision weights, quantized weights, and quantized model's intermediate activations, respectively. Despite the simplicity and stability, the weight-driven approach does not explicitly align with the primary objective of quantization, which is to preserve the full-precision model's final prediction. The output-driven approach is, therefore, more principled, as it minimizes the discrepancy in the model's outputs. However, naively applying output-driven objectives in the 1-bit regime leads to significant and often unexpected performance degradation. In this work, we identify that this failure arises from two key issues.

First, error accumulation across layers. In practice, existing binary LLM quantization frameworks (Li et al., 2025b) prioritize matching $\widehat{X}\widehat{W}$ with the intermediate representations $\widehat{X}W$ rather than the original full-precision signals XW with X being the full-precision model's activation. However, as quantization errors accumulate, these local targets progressively drift from the true model behavior, creating a discrepancy between layer-wise optimization and the global end-to-end objective. **Although recent work (Li et al., 2025a) addresses error propagation during the error-compensation stage after each quantization decision, the same accumulated input mismatch also affects the quantization-variable optimization itself, which remains largely unexplored.**

Second, magnitude bias and directional distortion in the optimization process. Simply minimizing mean squared error (MSE) in Euclidean space allocates the limited reconstruction capacity of a 1-bit model unevenly: dimensions with larger activation magnitude contribute more strongly to the objective and therefore receive disproportionately larger reconstruction improvements. More importantly, MSE measures only the magnitude of representation error and does not distinguish between perturbations that primarily change representation scale and those that change representation direction. This distinction is particularly important in pre-normalized Transformers, where subsequent computation repeatedly operates on normalized representations: normalization largely suppresses radial perturbations while preserving directional changes. Consequently, two quantization errors with similar MSE can have substantially different effects on the normalized representation. We find that unconstrained output alignment increasingly distorts the pairwise directional geometry among token representations, and that, even under matched reconstruction error, larger token-geometry distortion leads to larger downstream degradation.

Motivated by these observations, we propose a guided output-driven quantization method for 1-bit LLMs. Instead of relying on intermediate approximations, our method aligns quantized outputs with true full-precision targets, mitigating the impact of error accumulation. Furthermore, we formulate quantization as a geometry-constrained optimization problem that prevents reconstruction improvements from coming at the expense of token representation geometry. By constructing a local second-order approximation of the geometry constraint, we derive a closed-form projection that preserves token representation geometry while remaining close to the reconstruction-optimal update. These design choices yield a simple yet effective output-driven PTQ framework for 1-bit LLMs. **We focus on the binary setting, where the factorized binary parameterization admits efficient analytical conditional updates. Extending the proposed constrained formulation to multi-bit quantizers with discrete multi-level assignments is beyond the scope of this work.** In higher-bit quantization (e.g., 2-, 4-, or 8-bit), not all quantization variables admit analytic solutions; as a result, existing approaches in these settings typically rely on heuristic rounding schemes or gradient-based optimization which falls outside the scope of this work. In summary, our contributions are as follows:

- We identify two fundamental limitations of output-driven quantization in the 1-bit regime: *error accumulation* across layers and *anisotropic distortion* of token representations under MSE-based reconstruction. Through controlled perturbations with matched reconstruction error, we further show that larger token-geometry distortion leads to larger downstream degradation.
- We propose a novel PTQ method that optimizes true-target output reconstruction under a geometry-preserving constraint, mitigating the anisotropic distortion introduced by unconstrained output alignment. We further introduce a token-geometry-aware salient-mask selection strategy that identifies the most critical parameters for preserving token geometry.
- Extensive experiments demonstrate that our method consistently outperforms existing 1-bit PTQ techniques for LLMs.

2 RELATED WORKS

Quantization in LLMs. PTQ has emerged as a practical approach for compressing LLMs, as it can be applied directly to pretrained models with only a small calibration set, avoiding the high cost of Quantization-Aware Training (QAT). A range of PTQ methods have been proposed to improve the performance of quantized models further. For example, SmoothQuant (Xiao et al., 2023) mitigates activation outliers by applying scaling transformations that shift quantization difficulty from activations to weights. GPTQ (Frantar et al., 2023) formulates quantization as a two-stage process consisting of a quantization stage followed by an error compensation stage, where second-order Hessian information is exploited for layer-wise error correction. Building on this framework, subsequent works such as GPTAQ (Li et al., 2025a) and QEP (Arai & Ichikawa, 2025) aim to improve performance by addressing error propagation during the compensation stage. However, these methods primarily focus on mitigating error propagation after quantization, while paying relatively little attention to improving the initial quantization stage itself. More recent efforts such as QuIP# (Tseng et al., 2024) and QuaRot (Ashkboos et al., 2024) extend PTQ with rotation or vector quantization to better distribute outliers, though often at the expense of higher computational overhead. These efforts have helped LLMs maintain strong performance under moderate precision settings (e.g., 4-8 bits), yet the models still suffer from substantial degradation when pushed to extreme regimes. In this work, we focus on improving the binary quantization scheme through a closed-form formulation, without introducing additional inference computational overhead or auxiliary components.

1-Bit quantization for Large Language Models. Binarization, where weights are restricted to ± 1, represents the most aggressive form of quantization. It was first explored in computer vision with specialized binary architectures such as XNOR-Net (Rastegari et al., 2016) and Bi-Real Net (Liu et al., 2018), which showed that binary parameters could still capture meaningful representations. Follow-up studies (Guo et al., 2017; Xu et al., 2018) improved 1-bit quantization through enhanced coding schemes and optimized search strategies, enabling more accurate approximations of full-precision weights. Inspired by these advances, recent work has extended binarization to LLMs. Training-based approaches, such as BitNet (Wang et al., 2023), demonstrated that end-to-end training with binary weights is feasible. In contrast, post-training quantization (PTQ) approaches aim to binarize pretrained models with minimal retraining. BiLLM (Huang et al., 2024) selectively quantizes salient weights with low-bit precision while binarizing the rest, guided by Hessian-based importance and residual-aware masks. STB-LLM (Dong et al., 2024) combines pruning and quantization with fine-grained grouping, achieving sub-1-bit average precision while maintaining accuracy, albeit with added kernel and storage costs. Most recently, research has shifted toward data-aware and fine-grained quantizers tailored for 1-bit PTQ. ARB (Li et al., 2025b) introduces grouping and refinement strategies to reduce quantization error, and its data-aware extension ARB-X further optimizes the output alignment.

3 PRELIMINARY ANALYSIS

Output reconstruction improves MSE unevenly and distorts token geometry. We first examine why output-driven quantization can achieve lower reconstruction error yet inferior downstream performance in the extreme 1-bit regime. Given the full-precision output Y^{FP} and its quantized counterpart $\widehat{Y}$, output-driven methods minimize the difference between Y^{FP} and $\widehat{Y}$:

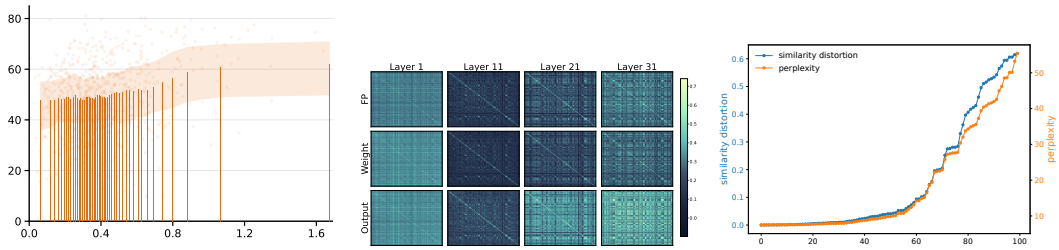
$$\mathcal{L}_{\mathrm{rec}} = \|Y^{\mathrm{FP}} - \widehat{Y}\|_F^2. \tag{1}$$

Because this objective is dominated by dimensions with large absolute errors, the limited reconstruction capacity of a 1-bit model need not be allocated uniformly across the representation. We quantify this effect using the fractional error reduction of each output channel c:

$$R_c = \frac{E_c^{\mathrm{before}} - E_c^{\mathrm{after}}}{E_c^{\mathrm{before}} + \epsilon}, \tag{2}$$

$E_c = \|Y_{:,c}^{\mathrm{FP}} - \widehat{Y}_{:,c}\|_F^2$ is the residual in channel c, and ϵ is a small positive number to avoid dividing by zero. As shown in Fig. 1a, high-magnitude channels receive substantially larger fractional error reductions, revealing a strongly uneven allocation of reconstruction improvement.

Importantly, this reduction in MSE does not preserve the relational structure between tokens. This can be observed via the token geometry using the normalized Gram matrix $G(Y) = \overline{Y}\,\overline{Y}^{\top}$ where



(a) Uneven reconstruction improvement. Output-driven optimization preferentially improves high-magnitude representation channels.

(b) Accumulated geometry distortion. Output-driven optimization progressively deviates from the full-precision token geometry across layers.

(c) Geometry matters under controlled reconstruction error. At identical perturbation magnitude, larger token-geometry distortion leads to larger degradation.

Figure 1: **Reconstruction error alone does not characterize the functional distortion introduced by extreme quantization.** (a) Output-driven optimization allocates its reconstruction improvement unevenly across representation dimensions, disproportionately benefiting high-magnitude channels. (b) Despite reducing reconstruction error, it induces increasing distortion of the pairwise geometry between token representations. (c) We isolate this effect by perturbing the first-block representations with equal-magnitude errors while varying only their direction. Although all perturbations have identical reconstruction error, those that induce larger token-geometry distortion lead to higher final perplexity. Together, these observations suggest that preserving representation direction and token relationships captures information missed by reconstruction MSE.

$\overline{Y}_{t,:} = \frac{Y_{t,:}}{\|Y_{t,:}\|_F + \epsilon}$ is the normalized output vector. Each entry in the Gram matrix $G(Y)$ is the pairwise cosine similarity between token representations. Fig. 1b shows that output-driven optimization progressively departs from the full-precision (FP) token geometry, even as reconstruction error decreases, whereas weight-driven quantization preserves this structure substantially better. The discrepancy further increases with depth, suggesting that the geometry distortion introduced by local reconstruction accumulates through the network.

Why is reconstruction error insufficient? Reconstruction MSE measures the total magnitude of a perturbation but does not distinguish between errors that change representation scale and those that change representation direction. For a quantization error δ, we can decompose

$$\delta = \delta_{\parallel} + \delta_{\perp}, \qquad \|\delta\|_2^2 = \|\delta_{\parallel}\|_2^2 + \|\delta_{\perp}\|_2^2, \tag{3}$$

where $\delta_{\parallel}$ is parallel to the original representation and $\delta_{\perp}$ changes its direction. Reconstruction MSE penalizes both components equally. This distinction is important in pre-normalized Transformers. Ignoring the learned gain and numerical stabilizer, RMSNorm maps each token approximately to $y/\|y\|_2$. It therefore largely suppresses scale-changing errors while preserving errors that alter representation direction. Consequently, two perturbations with identical reconstruction error can induce different normalized representations. The cosine geometry in $G(Y)$ directly captures these changes in relative token direction, providing information that reconstruction alone cannot distinguish.

Does token geometry matter beyond reconstruction error? To isolate the effect of perturbation direction from its magnitude, we inject 100 independently sampled perturbations into the output of the first Transformer block. Specifically, for each trial we sample E_i with i.i.d. Gaussian entries and construct $\Delta_i = \eta \frac{\|Y\|_F}{\|E_i\|_F} E_i, \quad i = 1, \ldots, 100$, such that $\frac{\|\Delta_i\|_F}{\|Y\|_F} = 0.5$ for all trials. We then continue the forward pass from $Y_i = Y + \Delta_i$ and measure both the resulting token-geometry distortion and final perplexity. Since all perturbations have identical Frobenius norm, their reconstruction errors are exactly matched, while their directions differ. As shown in Fig. 1c, larger token-geometry distortion is strongly associated with larger performance degradation even under matched reconstruction error.

4 METHOD

Our method uses token geometry as a common principle at two points in the quantization pipeline. During parameter refinement, it filters reconstruction-optimal updates through a geometry-preserving constraint. Before refinement, it uses the same representation-level criterion to allocate

the limited residual binary capacity. We retain error-aware output reconstruction as the optimization backbone and introduce the geometry terms only where reconstruction error alone is insufficient. Consider a linear layer with full-precision weights $W \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$. Let $X, \widehat{X} \in \mathbb{R}^{N \times d_{\text{in}}}$ denote the inputs generated by the full-precision and partially quantized models, respectively. We parameterize the binary weight as

$$\widehat{W} = \text{Diag}(\alpha_r)\, B\, \text{Diag}(\alpha_c), \tag{4}$$

where $B \in \{-1, +1\}^{d_{\text{in}} \times d_{\text{out}}}$, $\alpha_r \in \mathbb{R}^{d_{\text{in}}}$, and $\alpha_c \in \mathbb{R}^{d_{\text{out}}}$, and $\text{Diag}(v)$ denotes a diagonal matrix constructed from vector v.

4.1 Error-Aware Output Reconstruction

Pseudo-target reconstruction. Conventional layer-wise output alignment reconstructs the response of the full-precision weight to the already perturbed activation:

$$\mathcal{L}_{\text{pseudo}}(\widehat{W}) = \|\widehat{X}W - \widehat{X}\widehat{W}\|_F^2 = \|\widehat{X}(W - \widehat{W})\|_F^2. \tag{5}$$

Although this objective provides a local regression target, it does not explicitly account for the discrepancy between X and $\widehat{X}$. As quantization proceeds through the network, this discrepancy accumulates and progressively shifts the layer-wise target away from the full-precision target.

True-target reconstruction. We instead reconstruct the full-precision layer output:

$$\mathcal{L}_Q(\widehat{W}) = \|XW - \widehat{X}\widehat{W}\|_F^2. \tag{6}$$

The reconstruction loss in Eq. (6) can then be rewritten in terms of the current weight approximation error and the inherited activation residual as follows:

$$\begin{aligned} L_Q(\widehat{W}) &= \|\widehat{X}(W - \widehat{W}) + (X - \widehat{X})W\|_F^2 \\ &= \|\widehat{X}(W - \widehat{W})\|_F^2 + \|(X - \widehat{X})W\|_F^2 + 2\langle \widehat{X}(W - \widehat{W}), (X - \widehat{X})W\rangle_F, \end{aligned} \tag{7}$$

where $\langle U, V\rangle_F = \text{Tr}(U^\top V)$. The first term in Eq. (7) measures the error introduced by the current quantized layer, whereas the second term represents the error inherited from preceding layers and is constant with respect to $\widehat{W}$. For compactness, we define $\widehat{S} = \widehat{X}^\top \widehat{X}$ and $S = \widehat{X}^\top X$ to rewrite the reconstruction loss as follows:

$$\mathcal{L}_Q(\widehat{W}) = \text{Tr}(\widehat{W}^\top \widehat{S}\widehat{W}) - 2\,\text{Tr}(\widehat{W}^\top S W) + \text{Tr}(W^\top X^\top X W). \tag{8}$$

Although Eq. (8) is an unconstrained quadratic program w.r.t. $\widehat{W}$, optimizing it under the coupled factorization in Eq. (4) is a mixed discrete-continuous problem. We therefore optimize the quantization variables conditionally while holding the remaining variables unchanged.

4.2 Token-Geometry Regularization

Our analysis in Section 3 characterizes token geometry through the normalized pairwise similarities. Directly optimizing this quantity introduces representation-dependent normalization, breaking the blockwise quadratic structure required by our analytical binary updates. We therefore use the following cross-Gram affinity as a tractable surrogate:

$$\mathcal{S}_{\text{Reg}}(\widehat{W}) = {}^1\!/_2\langle (\widehat{X}\widehat{W})(\widehat{X}\widehat{W})^\top, (XW)(XW)^\top\rangle_F = {}^1\!/_2\,\text{Tr}\left(\widehat{W}^\top M \widehat{W}\right), \tag{9}$$

where $M = \widehat{X}^\top X W W^\top X^\top \widehat{X} = S W W^\top S^\top$. We use this cross-Gram affinity as a tractable surrogate for token geometry: it captures agreement between all pairs of token outputs while remaining quadratic in each block of quantization variables. In the Appendix A, we provide visualizations showing that optimizing this surrogate consistently improves the target geometry measures. A larger $\mathcal{S}_{\text{Reg}}$ indicates stronger affinity between the pairwise inner-product structures of the full-precision and quantized token representations. Accordingly, we optimize the reconstruction objective while requiring the geometry-alignment score not to decrease:

$$\min_{\alpha_r, \alpha_c, B} \mathcal{L}_Q(\widehat{W}) \quad \text{s.t.: } \mathcal{S}_{\text{Reg}}(\widehat{W}) \geq \mathcal{S}_{\text{Reg}}(\widehat{W}^{(t)}), \tag{10}$$

where $\widehat{W}^{(t)}$ denotes the current quantized weights. The reconstruction term controls output scale, while the constraint prevents an accepted update from improving reconstruction by sacrificing relational structure. Thus, the reconstruction objective determines the preferred update, while the geometry constraint restricts updates that distort token relationships.

The joint problem contains continuous scales, discrete binary variables, and a coupled quadratic constraint, hence it has no joint closed-form solution. We optimize one variable block $\phi \in \{\alpha_c, \alpha_r, B_{i,:}\}$ at a time, while keeping all remaining variables fixed.

4.3 Reconstruction-Optimal Proposals

For a current variable block ϕ, we first compute its reconstruction closed-form optimum for $\mathcal{L}_Q$ as $\phi_Q^\star \in \arg\min_\phi \mathcal{L}_Q(\phi)$ (see Appendix D). The update $\Delta_\phi^Q = \phi_Q^\star - \phi$ is the analytical closed-form solution derived by minimizing $\mathcal{L}_Q$ while keeping other variables fixed. This update step may violate the constraint in Eq. (10). We therefore seek a feasible update that remains close to the reconstruction optimum. To retain the analytical nature of the proposal, each coordinate either takes its complete proposed change or remains unchanged:

$$\min_\Delta \quad {}^1\!/_2\|\Delta - \Delta_\phi^Q\|^2, \quad \text{s.t.: } \begin{cases} \mathcal{S}_{\text{Reg}}(\phi+\Delta) \geq \mathcal{S}_{\text{Reg}}(\phi), \\ \Delta_j \in \{0, (\Delta_\phi^Q)_j\}, \forall j. \end{cases} \tag{11}$$

4.4 Local Second-Order Geometry Constraint

The exact quadratic geometry constraint in Eq. (11) couples the coordinates of Δ, making the discrete projection intractable. We obtain a separable local surrogate by expanding $\mathcal{S}_{\text{Reg}}$ around the current variable block ϕ and retaining only diagonal curvature:

$$\mathcal{S}_{\text{Reg}}(\phi+\Delta) - \mathcal{S}_{\text{Reg}}(\phi) = g_\phi^\top \Delta + {}^1\!/_2 \Delta^\top H_\phi \Delta \approx \textstyle\sum_j \left[(g_\phi)_j \Delta_j + {}^\beta\!/_2 (h_\phi)_j \Delta_j^2\right], \tag{12}$$

where $g_\phi = \nabla_\phi \mathcal{S}_{\text{Reg}}(\phi)$ is the gradient, $H_\phi = \nabla_\phi^2 \mathcal{S}_{\text{Reg}}(\phi)$ is the Hessian matrix, $h_\phi = \text{diag}(H_\phi)$ is the Hessian diagonal. The coefficient $\beta \geq 0$ controls the strength of this term. To obtain a separable feasible region, we impose the local constraint in Eq. (12) on every coordinate:

$$(g_\phi)_j \Delta_j + {}^\beta\!/_2 (h_\phi)_j \Delta_j^2 \geq 0, \forall j. \tag{13}$$

Substituting this approximation into Eq. (10) gives:

$$\min_\Delta \ {}^1\!/_2\|\Delta - \Delta_\phi^Q\|^2 \quad \text{s.t.: } \begin{cases} (g_\phi)_j \Delta_j + {}^\beta\!/_2 (h_\phi)_j \Delta_j^2 \geq 0, \forall j, \\ \Delta_j \in \left\{0, (\Delta_\phi^Q)_j\right\}, \forall j. \end{cases} \tag{14}$$

4.5 Closed-Form Geometry-Preserving Projection

For each coordinate j in Eq. (14), retaining $(\Delta_\phi^Q)_j$ produces zero projection error and is optimal whenever it satisfies the local geometry constraint. Otherwise, the only remaining feasible choice is zero. We therefore define

$$(M_\phi)_j = \mathbf{1}\left[(g_\phi)_j (\Delta_\phi^Q)_j + {}^\beta\!/_2 (h_\phi)_j (\Delta_\phi^Q)_j^2 \geq 0\right], \tag{15}$$

where $\mathbf{1}(.) \in \{0,1\}$ is the indicator function. The projected update is

$$\widetilde{\Delta}_\phi = M_\phi \odot \Delta_\phi^Q, \qquad \widetilde{\phi} = \phi + \widetilde{\Delta}_\phi. \tag{16}$$

Proposition 1 states the local property for constructing the candidate update (see Appendix C for the proof).

Proposition 1 (Local geometry preservation)**.** *Under the diagonal-curvature approximation in Eq.* (12)*, the projected update does not decrease the local geometry-alignment model:*

$$\textstyle\sum_j \left[(g_\phi)_j \widetilde{\Delta}_{\phi,j} + \frac{\beta}{2} (h_\phi)_j \widetilde{\Delta}_{\phi,j}^2\right] \geq 0.$$

The update in Eq. (16) is the exact solution of the *discrete separable local projection* in Eq. (14). Because diagonalization is an approximation, we accept the candidate only when the actual reconstruction loss does not increase and the actual score $\mathcal{S}_{\text{Reg}}$ does not decrease. This final check enforces the original constraint in Eq. (11).

4.6 Conditional Reconstruction Solutions

Our optimization alternates over each quantization variable $\phi \in \{\alpha_r, \alpha_c, B\}$ while keeping the remaining variables fixed. Each subproblem admits an analytical closed-form update, which is used to compute Δ_ϕ^Q. We also provide the geometry terms, g_ϕ and h_ϕ, used by the proposed constrained optimization. The complete derivations are provided in Appendices B and D.

4.7 Token-Geometry-Guided Mask Selection

For each layer, the binary quantizer uses two binary matrices to represent a subset of salient columns and one binary matrix for the remaining columns. Rather than determining this subset solely from weight reconstruction error, we use token geometry to determine the final allocation. Following the saliency criterion of the underlying binary quantizer, we define the full-precision activation Gram matrix $H = X^\top X + \epsilon \mathbf{I}$ and let U denote the upper-triangular Cholesky factor satisfying $H^{-1} = U^\top U$. We rank the d_{in} input columns according to

$$s_j = \textstyle\sum_i (W_{ij})^2 / U_{jj}^2, \tag{17}$$

and denote the resulting descending order by $\{c_1, \ldots, c_{d_{\text{in}}}\}$. For each quantization group and each prefix $\mathcal{C}_m = \{c_1, \ldots, c_m\}$, $1 \le m < K$, let $\widetilde{W}_q^{(m)}$ denote the pseudo-quantized weight obtained when the columns in $\mathcal{C}_m$ receive the salient two-term representation. Among these $K-1$ candidates, we select the mask whose induced token geometry is closest to that of the full-precision output:

$$m^\star = \arg\min_{1 \le m < K} \frac{\left\| Z_q^{(m)} Z_q^{(m)\top} - Z Z^\top \right\|_F^2}{\|Z Z^\top\|_F^2 + \epsilon}. \tag{18}$$

Here, Z and $Z_q^{(m)}$ are respectively the FP output and the pseudo-quantized output after projection to a lower-dimensional space. Specifically, we use $P \in \mathbb{R}^{d_{\text{out}} \times r}$ with orthonormal columns, where $r \ll d_{\text{out}}$, and construct P by sampling a Gaussian random matrix and orthonormalizing its columns using reduced QR decomposition. The same projection is used for all candidate masks within the group. The selected prefix $\mathcal{C}_{m^\star}$ defines the salient columns. Since $m^\star$ is selected independently for each group, K bounds the candidate range and is chosen such that the resulting average bit-width remains comparable to the underlying baseline. This converts mask selection into an explicit token-geometry preservation problem.

4.8 Overall Algorithm

We initialize the binary weights with the signs of the full-precision weights, as in conventional binary quantization. We apply output-driven refinement to the final fully connected layer of each Transformer block, with the remaining linear layers following ARB-RC. We then select the two-term residual mask using Section 4.7 and iteratively refine the quantization parameters using the geometry-preserving projection. Thus, token geometry governs both capacity allocation and update acceptance. The complete procedure for iterative refinement is provided in Appendix F.

5 Experiments

Models and datasets. Our experiments are conducted on the OPT model family (Zhang et al., 2022) as well as on the LLaMA family, covering parameter scales from 1.3B to 13B, including LLaMA-2 (Touvron et al., 2023) and LLaMA-3 (Grattafiori et al., 2024). We also report results on newer architectures from the Qwen family. For evaluation, *perplexity* is reported on WikiText2 (Merity et al., 2016), C4 (Raffel et al., 2020) and PTB (Marcus et al., 1993), which are standard for measuring language modeling quality. To assess downstream capability, we also measure zero-shot performance on eight QA reasoning datasets: ARC-Easy and ARC-Challenge (Clark et al., 2018), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021) and OBQA (Mihaylov et al., 2018), COPA(Gordon et al., 2012) and LAMBADA (Paperno et al., 2016).

Experimental setting. All experiments are implemented in PyTorch and executed on a single NVIDIA A100 GPU. To be consistent with prior studies such as GPTQ (Frantar et al., 2023) and BiLLM (Huang et al., 2024), we use 128 samples from the C4 dataset with a sequence length of 2048 as calibration data to enable fair comparison. The quantization block size is fixed at 128, and the optimization of quantization variables is performed with 15 iterations to be consistent with ARB

Table 1: Perplexity (↓) results of OPT and LLaMA models on C4, WikiText2, and PTB. Separate weight-bit columns are used for each model family. **Use OA** denotes methods that use an output-alignment objective for quantization.

Dataset	Method	Use OA	OPT Bits	OPT 1.3B	OPT 2.7B	OPT 6.7B	LLaMA Bits	LLaMA LLaMA-2 7B	LLaMA LLaMA-2 13B	LLaMA LLaMA-3 8B
C4	Full Precision	–	16	16.07	14.34	12.71	16	7.26	6.73	9.45
	PB-LLM	×	1.7	168.12	222.15	104.78	1.7	80.69	184.67	104.15
	BiLLM	×	1.11	64.14	44.77	42.13	1.08	39.38	25.87	61.04
	ARB-X	✓	1.11	47.60	34.97	22.54	1.08	28.02	19.82	41.86
	OAC	×	1.11	–	–	–	1.08	21.64	15.25	–
	DBellQuant	×	1.11	42.86	33.19	21.75	1.08	21.83	15.14	–
	ARB-RC	×	1.11	28.03 ± 0.30	21.51 ± 0.15	17.02 ± 0.10	1.08	20.56 ± 0.68	14.76 ± 0.50	36.04 ± 0.01
	Ours	✓	1.11	$\mathbf{24.65 \pm 0.05}$	$\mathbf{19.90 \pm 0.06}$	$\mathbf{16.18 \pm 0.03}$	1.08	$\mathbf{17.74 \pm 0.30}$	$\mathbf{13.69 \pm 0.11}$	$\mathbf{35.16 \pm 0.20}$
WikiText2	Full Precision	–	16	14.62	12.47	10.86	16	5.47	4.88	6.14
	PB-LLM	×	1.7	239.81	278.27	144.25	1.7	66.41	236.40	73.08
	BiLLM	×	1.11	69.05	48.61	47.65	1.08	32.31	21.35	55.80
	ARB-X	✓	1.11	45.40	34.37	20.07	1.08	21.61	14.86	31.98
	OAC	×	1.11	–	–	–	1.08	19.50	13.15	–
	DBellQuant	×	1.11	43.42	31.47	18.89	1.08	17.91	12.79	–
	ARB-RC	×	1.11	26.83 ± 0.15	20.09 ± 0.19	15.06 ± 0.08	1.08	16.43 ± 0.06	12.33 ± 0.15	28.14 ± 0.55
	Ours	✓	1.11	$\mathbf{24.11 \pm 0.25}$	$\mathbf{18.22 \pm 0.03}$	$\mathbf{14.28 \pm 0.18}$	1.08	$\mathbf{14.62 \pm 0.05}$	$\mathbf{10.98 \pm 0.19}$	$\mathbf{26.62 \pm 0.29}$
PTB	Full Precision	–	16	20.29	17.97	15.77	16	37.91	50.93	11.18
	PB-LLM	×	1.7	324.62	183.97	169.49	1.7	657.24	816.31	106.25
	BiLLM	×	1.11	115.94	88.52	69.41	1.08	5243.01	309.12	87.25
	ARB-X	✓	1.11	71.96	54.28	31.23	1.08	783.66	208.06	62.69
	ARB-RC	×	1.11	42.91 ± 0.20	31.20 ± 0.07	22.21 ± 0.11	1.08	853.30 ± 196.57	196.31 ± 2.05	47.86 ± 1.47
	Ours	✓	1.11	$\mathbf{37.03 \pm 0.40}$	$\mathbf{28.69 \pm 0.51}$	$\mathbf{21.24 \pm 0.42}$	1.08	$\mathbf{248.56 \pm 45.39}$	$\mathbf{188.54 \pm 1.6}$	$\mathbf{40.16 \pm 1.3}$

Table 2: Zero-shot accuracy (%) on QA reasoning benchmarks. Green values denote the improvement of our method over ARB-RC in percentage points. †The average is computed over all tasks.

Model	Method	Bits	Lambada	PIQA	OBQA	WinoGrande	ARC-E	ARC-C	HellaSwag	COPA	Avg.†
LLaMA-2-7B	ARB-RC	1.08	51.66	64.85	19.40	59.51	50.97	24.32	36.79	71.00	47.31
	DBELL	1.08	41.63	64.85	17.60	56.91	50.38	23.98	35.65	63.00	44.25
	Ours	1.08	**58.53** (+6.87)	**67.46** (+2.61)	**20.80** (+1.40)	**60.14** (+0.63)	**53.41** (+2.44)	**25.60** (+1.28)	**38.91** (+2.12)	**74.00** (+3.00)	**49.86** (+2.55)
LLaMA-2-13B	ARB-RC	1.08	66.02	69.59	23.20	61.48	61.28	29.18	38.73	81.00	53.81
	DBELL	1.08	55.75	68.44	20.80	61.64	60.10	28.16	39.08	78.00	51.50
	Ours	1.08	**69.28** (+3.26)	**71.71** (+2.12)	**25.00** (+1.80)	**66.22** (+4.74)	**64.44** (+3.16)	**30.89** (+1.71)	**43.09** (+4.36)	**82.00** (+1.00)	**56.58** (+2.77)
LLaMA-3-8B	ARB-RC	1.08	48.85	61.43	20.20	57.46	45.83	22.01	35.39	76.00	45.90
	DBELL	1.08	34.78	59.30	17.80	54.85	36.78	18.86	33.40	73.00	41.10
	Ours	1.08	**57.73** (+8.88)	**67.19** (+5.76)	**24.60** (+4.40)	**59.35** (+1.89)	**56.82** (+10.99)	**26.62** (+4.61)	**38.40** (+3.01)	**77.00** (+1.00)	**50.96** (+5.06)

(Li et al., 2025b), β is set to 1 for simplicity. Following the baselines, we only quantize the model weights, keeping the activation at full-precision. For token-geometry-aware mask selection, we set $K = 25$ for Llama and Qwen while set $K = 30$ for OPT, and set $r = 32$ for all experiments.

Baseline methods. We compare our method against several state-of-the-art 1-bit PTQ methods, including BiLLM (Huang et al., 2024), ARB-LLM(Li et al., 2025b) and PB-LLM (Shang et al., 2023), ensuring that all implementations adhere to the details provided in their respective papers. BiLLM (Huang et al., 2024), ARB-LLM(Li et al., 2025b), PB-LLM (Shang et al., 2023) and DBellQuant (Ye et al., 2025) all utilize the PTQ approach for model calibration of GPTQ. For controlled comparison, we primarily consider 1-bit PTQ methods that retain the standard binary-weight representation without introducing additional representation transforms. ARB-RC results and our results in Tables 1 were obtained by running with 3 different random seeds.

5.1 Experimental results

Results on OPT and LLaMA models. As shown in Table 1, our method consistently outperforms ARB-RC across all 18 perplexity comparisons on OPT and LLaMA models. The improvement is particularly pronounced in challenging settings, reducing PTB perplexity from 853.30 to 248.56 on LLaMA-2-7B and from 47.86 to 40.16 on LLaMA-3-8B. For downstream reasoning tasks, our method improves the average zero-shot accuracy over ARB-RC by 2.55, 2.77, and 5.06 % points on LLaMA-2-7B, LLaMA-2-13B, and LLaMA-3-8B, respectively, and outperforms ARB-RC on all 24 individual benchmarks. The largest gain reaches 10.99 % in accuracy on ARC-E for LLaMA-3-8B, demonstrating the effectiveness of the method.

Performance of our method over Qwen models. We further evaluate Qwen2, Qwen2.5, and Qwen3 to examine whether the benefit of geometry preservation transfers across model architectures. As shown in Table 3, our method improves over ARB-RC in all 15 perplexity comparisons

Table 3: Perplexity (↓) on C4, WikiText2, and PTB, and average zero-shot reasoning accuracy (↑) for Qwen models. All methods use a block size of 128. Green annotations report the absolute improvement over ARB-RC.

Dataset	Metric	Method	Block Size	Qwen2		Qwen2.5		Qwen3
				1.5B	7B	1.5B	7B	8B
C4	PPL (↓)	ARB-RC	128	109.83	22.63	128.77	24.32	32.71
		DBellQuant	128	607.06	29.03	1428.88	29.52	50.88
		Ours	128	**74.83** (↓35.00)	**20.54** (↓2.09)	**64.33** (↓64.44)	**20.95** (↓3.37)	**27.68** (↓5.03)
WikiText2	PPL (↓)	ARB-RC	128	89.59	16.99	107.92	17.08	26.03
		DBellQuant	128	542.06	28.52	2416.90	25.10	58.16
		Ours	128	**68.45** (↓21.14)	**14.55** (↓2.44)	**48.11** (↓59.81)	**14.42** (↓2.66)	**20.73** (↓5.30)
PTB	PPL (↓)	ARB-RC	128	135.49	28.33	175.40	29.47	44.56
		DBellQuant	128	1246.88	47.27	3843.26	43.99	86.66
		Ours	128	**107.04** (↓28.45)	**23.50** (↓4.83)	**96.48** (↓78.92)	**25.24** (↓4.23)	**33.22** (↓11.34)
AveQA	Acc. (↑)	ARB-RC	128	35.62	53.95	37.83	54.79	53.56
		DBellQuant	128	31.53	47.01	32.17	49.02	43.84
		Ours	128	**38.32** (+2.70)	**56.03** (+2.08)	**41.62** (+3.79)	**55.89** (+1.10)	**55.22** (+1.65)

and all five average QA comparisons. The gains are substantial on Qwen2.5-1.5B, where perplexity decreases from 128.77 to 64.33 on C4, from 107.92 to 48.11 on WikiText2, and from 175.40 to 96.48 on PTB, while average QA accuracy improves by 3.79 percentage points. These results show that preserving token geometry provides consistent benefits beyond a single model family.

5.2 Ablation Study

Ablation of the constrained optimization objective.

Table 4 ablates true-target reconstruction $\mathcal{L}_Q$, the geometry-preservation constraint, and the geometry-guided mask selection. On Qwen2.5-1.5B, replacing the pseudo target with $\mathcal{L}_Q$ reduces perplexity from 128.77 to 115.69 on C4 and from 107.92 to 98.18 on WikiText2. In contrast, on LLaMA-2-7B, the same change substantially degrades performance, increasing perplexity from 20.56 to 29.12 and from 16.36 to 26.24, respectively. This result shows that output reconstruction alone does not reliably preserve important representation structure, whereas the geometry constraint stabilizes the reconstruction update across both architectures. Combined with $\mathcal{L}_Q$, it improves the baseline by 19.24 and 17.34 perplexity points on Qwen2.5-1.5B, and by 1.62 and 1.16 points on LLaMA-2-7B, over C4 and WikiText2, respectively. Overall, the complete method improves the corresponding baseline by 64.44 and 59.81 points on Qwen2.5-1.5B, and by 2.82 and 1.74 points on LLaMA-2-7B.

Table 4: Ablation study about the contribution of each component, using a fixed random seed.

Optimization			Qwen-2.5-1.5B		LLaMA-2-7B	
$\mathcal{L}_Q$	$\Delta\mathcal{S}_{\text{Reg}} \geq 0$	Mask	C4	Wiki	C4	Wiki
×	×	×	128.77	107.92	20.56	16.36
✓	×	×	115.69	98.18	29.12	26.24
×	✓	×	119.14	99.45	19.97	15.66
×	×	✓	86.00	73.07	19.91	16.17
✓	✓	×	109.53	90.58	18.94	15.20
✓	✓	✓	**64.33**	**48.11**	**17.74**	**14.62**

Inference overhead. Our method introduces no additional inference-time computation relative to ARB-RC. Both the geometry-preserving optimization and the geometry-guided mask selection are performed only during offline quantization, while the resulting quantized model retains the same inference representation and forward-pass computation as ARB-RC. We further report measured prefill and decoding throughput in Table 6, confirming comparable inference efficiency.

6 Conclusion

In this work, we revisit output-driven optimization for 1-bit post-training quantization of large language models. We show that conventional output reconstruction can distort representation geometry through anisotropic optimization, and address this by formulating quantization as constrained reconstruction with geometry preservation. Experiments across multiple LLM families and benchmarks show consistent improvements in perplexity and downstream accuracy over strong 1-bit PTQ baselines.

AI USE STATEMENT

In this work, we used generative AI tools to assist with aspects of implementation and language editing, including improving the clarity and presentation of the manuscript and suggesting improvements to scientific visualizations and figure presentation. All AI-assisted code was reviewed and tested by the authors, and all AI-assisted text and visualization suggestions were reviewed and revised to ensure that they accurately reflect the underlying methods and experimental results. We did not rely on generative AI to generate experimental results or make final methodological decisions. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REFERENCES

Yamato Arai and Yuma Ichikawa. Quantization error propagation: Revisiting layer-wise post-training quantization. *ArXiv*, abs/2504.09629, 2025. URL https://api.semanticscholar.org/CorpusID:277781512.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. *NeurIPS*, 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(05), pp. 7432–7439, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jungwook Choi, Pierce I-Jen Chuang, Zhuo Wang, Swagath Venkataramani, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Bridging the accuracy gap for 2-bit quantized neural networks (qnn). *arXiv preprint arXiv:1807.06964*, 2018.

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*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusID:52967399.

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

Peijie Dong, Lujun Li, Dayou Du, Yuhan Chen, Zhenheng Tang, Qiang Wang, Wei Xue, Wenhan Luo, Qi fei Liu, Yi-Ting Guo, and Xiaowen Chu. Stbllm: Breaking the 1-bit barrier with structured binary llms. *ArXiv*, abs/2408.01803, 2024. URL https://api.semanticscholar.org/CorpusID:271710591.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Çelebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22:107:1–107:48, 2020. URL https://api.semanticscholar.org/CorpusID:224814118.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *ICLR*, 2023.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In Eneko Agirre, Johan Bos, Mona Diab, Suresh Manandhar, Yuval Marton, and Deniz Yuret (eds.), **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main*

conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), pp. 394–398, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL https://aclanthology.org/S12-1052/.

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.

Yiwen Guo, Anbang Yao, Hao Zhao, and Yurong Chen. Network sketching: Exploiting binary structure in deep cnns. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4040–4048, 2017. URL https://api.semanticscholar.org/CorpusID:11244259.

Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Cornell University - arXiv,Cornell University - arXiv*, Sep 2020.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Wei Huang, Yangdong Liu, Haotong Qin, Ying Li, Shiming Zhang, Xianglong Liu, Michele Magno, and Xiaojuan Qi. Billm: Pushing the limit of post-training quantization for llms. *arXiv preprint arXiv:2402.04291*, 2024.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam M. Shazeer, and Z. Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *ArXiv*, abs/2006.16668, 2020. URL https://api.semanticscholar.org/CorpusID:220265858.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdel rahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusID:204960716.

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. 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. 36690–36706. PMLR, 13–19 Jul 2025a. URL https://proceedings.mlr.press/v267/li25dn.html.

Zhiteng Li, Xianglong Yan, Tianao Zhang, Haotong Qin, Dong Xie, Jiang Tian, Zhongchao Shi, Linghe Kong, Yulun Zhang, and Xiaokang Yang. Arb-llm: Alternating refined binarizations for large language models. *ICLR*, abs/2410.03129, 2025b. URL https://api.semanticscholar.org/CorpusID:273163233.

Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, W. Liu, and K. Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. *ArXiv*, abs/1808.00278, 2018. URL https://api.semanticscholar.org/CorpusID:51892264.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL https://aclanthology.org/J93-2004/.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*, 2016.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, 2016. URL https://api.semanticscholar.org/CorpusID:14925907.

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.

Yuzhang Shang, Zhihang Yuan, Qiang Wu, and Zhen Dong. Pb-llm: Partially binarized large language models. *arXiv*, 2023.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, et al. Flatquant: Flatness matters for llm quantization. In *ICML*, 2025.

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*, 2023.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks. *ICML*, 2024.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. Bitnet: Scaling 1-bit transformers for large language models. *ArXiv*, abs/2310.11453, 2023. URL https://api.semanticscholar.org/CorpusID:264172438.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, EdH. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, Jun 2022.

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 *ICML*, 2023.

Chen Xu, Jianqiang Yao, Zhouchen Lin, Wenwu Ou, Yuanbin Cao, Zhirong Wang, and Hongbin Zha. Alternating multi-bit quantization for recurrent neural networks. In *International Conference on Learning Representations*, volume abs/1802.00150, 2018. URL https://api.semanticscholar.org/CorpusID:8257350.

Zijian Ye, Wei Huang, Yifei Yu, Tianhe Ren, Zhongrui Wang, and Xiaojuan Qi. Dbellquant: Breaking the bell with double-bell transformation for llms post training binarization. *ArXiv*, abs/2507.01027, 2025. URL https://api.semanticscholar.org/CorpusID:280080895.

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

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *ArXiv*, abs/1912.08777, 2019. URL https://api.semanticscholar.org/CorpusID:209405420.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

A Impact Of Token Geometry Preservation On Attention Mechanism

Effect on true geometry-preserving target. As shown in Fig. 2, unconstrained output optimization progressively distorts the normalized token-similarity structure, whereas our geometry-constrained updates substantially preserve it across layers. This confirms that the tractable surrogate in $\mathcal{S}_{Reg}$ effectively protects the normalized geometry.

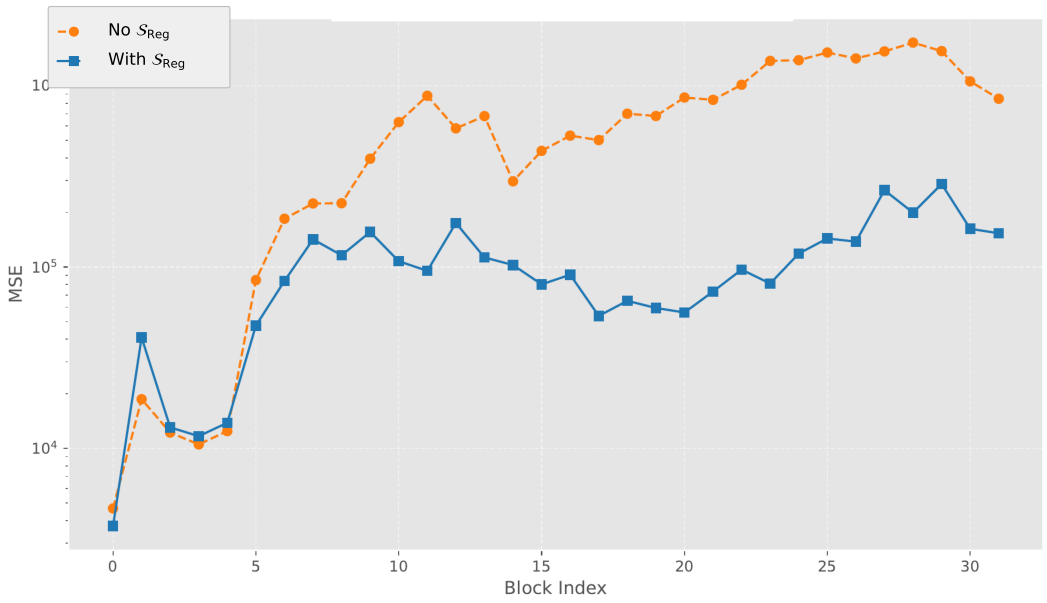


Figure 2: The difference between the token similarity matrix between full-precision model and quantized model, with and without $\mathcal{S}_{Reg}$ constraint, demonstrating the effectiveness of the proposed method.

Effect of geometry-preserving projection. As complementary qualitative evidence, we inspect whether geometry-preserving updates also better retain full-precision attention patterns. Figure 3 qualitatively compares the attention-score patterns produced with and without the geometry-preserving projection. Although the proposed constraint is defined on token-output geometry rather than directly on the attention matrix, the constrained update produces attention patterns that more closely resemble those of the full-precision model.

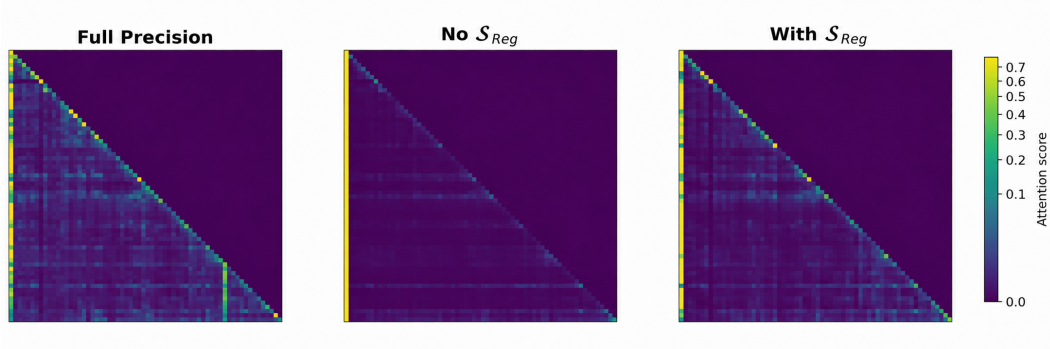


Figure 3: Comparison of attention scores with and without the token-geometry preservation constraint against the full-precision model. Attention scores produced with the $\mathcal{S}_{Reg}$ constraint more closely match those of the pretrained model than those obtained without $\mathcal{S}_{Reg}$ constraint, demonstrating the effectiveness of the proposed method.

B Analytical Solutions for Quantization Variables and Geometry Gradient

We provide the detailed derivations of the analytical closed-form solutions for the quantization variables α_r, α_c, and B.

Column scales. For fixed α_r and B, setting $\nabla_{\alpha_c}\mathcal{L}_Q = 0$ gives

$$\alpha_c^\star = \frac{\operatorname{diag}\left(B^\top \operatorname{Diag}(\alpha_r) S W\right)}{\operatorname{diag}\left(B^\top \operatorname{Diag}(\alpha_r)\widehat{S}\operatorname{Diag}(\alpha_r) B\right) + \varepsilon}, \tag{19}$$

where $\operatorname{diag}(.)$ denotes the diagonal vector of a matrix, and the division is element-wise. The corresponding update is $\Delta^Q_{\alpha_c} = \alpha_c^\star - \alpha_c$.

Row scales. For fixed α_c and B, define $K = \operatorname{Diag}(\alpha_c)^2, R = BKB^\top$. The stationary condition for α_r is

$$(\widehat{S} \odot R)\alpha_r = \operatorname{diag}\left(SW \operatorname{Diag}(\alpha_c) B^\top\right). \tag{20}$$

The closed-form solution is

$$\alpha_r^\star = (\widehat{S} \odot R)^\dagger \operatorname{diag}\left(SW \operatorname{Diag}(\alpha_c) B^\top\right), \tag{21}$$

where $(\cdot)^\dagger$ denotes the Moore–Penrose pseudoinverse. In implementation, we solve Eq. (20) using a numerically stable least-squares solver. The update is $\Delta^Q_{\alpha_r} = \alpha_r^\star - \alpha_r$.

Binary rows. For fixed α_r and α_c, define $N = \operatorname{Diag}(\alpha_r)\widehat{S}\operatorname{Diag}(\alpha_r)$, $K = \operatorname{Diag}(\alpha_c)^2$, $P = \operatorname{Diag}(\alpha_r) SW \operatorname{Diag}(\alpha_c)$. We can then derive the closed-form solution as:

$$B^\star_{i,:} = \operatorname{sign}(P - [N - \operatorname{diag}(\operatorname{Diag}(N))]BK)_{i,:}. \tag{22}$$

The binary update is therefore $\Delta^Q_{B_{i,:}} = B^\star_{i,:} - B_{i,:}$. Detailed derivations of Eqs. 19, 21, and 22 are provided in Appendix D.

B.1 Geometry Gradient and Diagonal Curvature

Let $G_{\widehat{W}} = \nabla_{\widehat{W}}\mathcal{S}_{Reg}(\widehat{W})$. Applying the chain rule to the parameterization in Eq. (4) gives

$$\begin{aligned} g_{\alpha_r} &= \operatorname{diag}\left(G_{\widehat{W}} \operatorname{Diag}(\alpha_c) B^\top\right), \\ g_{\alpha_c} &= \operatorname{diag}\left(B^\top \operatorname{Diag}(\alpha_r) G_{\widehat{W}}\right), \\ g_B &= \operatorname{Diag}(\alpha_r) G_{\widehat{W}} \operatorname{Diag}(\alpha_c). \end{aligned} \tag{23}$$

For a binary-row update, $g_{B_{i,:}}$ is the i-th row of g_B. The corresponding diagonal curvature vectors are

$$\begin{aligned} h_{\alpha_r} &= \operatorname{diag}(M) \odot \operatorname{diag}\left(BD_c^2 B^\top\right), \\ h_{\alpha_c} &= \operatorname{diag}\left(B^\top D_r M D_r B\right), \\ h_B &= \left(\alpha_r^{\odot 2} \odot \operatorname{diag}(M)\right)\left(\alpha_c^{\odot 2}\right)^\top. \end{aligned} \tag{24}$$

The gradient and diagonal curvature are substituted directly into Eq. (15).

C Proof of Proposition 1

Proof. Since $\mathcal{S}_{\mathrm{Reg}}$ is quadratic in each conditional variable block ϕ, its change around the current point is

$$\mathcal{S}_{\mathrm{Reg}}(\phi + \Delta) - \mathcal{S}_{\mathrm{Reg}}(\phi) = g_\phi^\top \Delta + \frac{1}{2}\Delta^\top H_\phi \Delta.$$

As introduced in Eq. 12, we approximate this coupled quadratic change using the separable diagonal-curvature model

$$\widetilde{\Delta\mathcal{S}}_{\mathrm{Reg}}(\Delta) = \sum_j \left[(g_\phi)_j \Delta_j + \frac{\beta}{2}(h_\phi)_j \Delta_j^2\right].$$

For every retained coordinate, $\Delta_j = (\Delta_\phi^Q)_j$, and the corresponding term is nonnegative by construction of M_ϕ. For every rejected coordinate, $\Delta_j = 0$, and the corresponding term is zero. Therefore, every term in the sum is nonnegative, and hence

$$\widetilde{\Delta\mathcal{S}}_{\mathrm{Reg}}(\widetilde{\Delta}_\phi) = \sum_j \left[(g_\phi)_j \widetilde{\Delta}_{\phi,j} + \frac{\beta}{2} (h_\phi)_j \widetilde{\Delta}_{\phi,j}^2 \right] \geq 0.$$

Thus, $\widetilde{\Delta}_\phi$ satisfies the separable local geometry constraint in Eq. 12. □

D Derivation of closed-forms

Derivation for α_c The gradient of the $\mathcal{L}_Q$ w.r.t α_c is:

$$\begin{aligned}
\frac{\partial \mathcal{L}_Q}{\partial \mathrm{diag}(\alpha_c)} &= \frac{\partial \mathcal{L}_Q}{\partial (XW - \widehat{X}\widehat{W})} \frac{\partial (XW - \widehat{X}\widehat{W})}{\partial \alpha_c} \\
&= -2B^\top \mathrm{diag}(\alpha_r)\widehat{X}^\top \times (XW - \widehat{X}\widehat{W}) \\
&= -2B^\top \mathrm{diag}(\alpha_r)\widehat{X}^\top \times (XW - \widehat{X}\mathrm{diag}(\alpha_r)B\mathrm{diag}(\alpha_c)).
\end{aligned} \tag{25}$$

Setting the diagonal of this gradient to 0, we have:

$$\begin{aligned}
&\mathrm{Diag}(B^\top \mathrm{diag}(\alpha_r)\widehat{S}\mathrm{diag}(\alpha_r)B) \odot \alpha_c = \mathrm{Diag}(B^\top \mathrm{diag}(\alpha_r)SW) \\
\implies &\alpha_c = \frac{\mathrm{Diag}(B^T \mathrm{diag}(\alpha_r)SW)}{\mathrm{Diag}(B^T \mathrm{diag}(\alpha_r)\widehat{S}\mathrm{diag}(\alpha_r)B)}.
\end{aligned} \tag{26}$$

Derivation for α_r The gradient of the objective w.r.t α_r is:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathrm{diag}(\alpha_r)} &= \frac{\partial \mathcal{L}}{\partial (XW - \widehat{X}\widehat{W})} \frac{\partial (XW - \widehat{X}\widehat{W})}{\partial \alpha_r} \\
&= -2\widehat{X}^\top (XW - \widehat{X}\widehat{W})\mathrm{diag}(\alpha_c)B^\top \\
&= -2\widehat{X}^\top (XW - \widehat{X}\mathrm{diag}(\alpha_r)B\mathrm{diag}(\alpha_c))\mathrm{diag}(\alpha_c)B^\top
\end{aligned} \tag{27}$$

With $\widehat{S} = \widehat{X}^\top \widehat{X}$, setting the diagonal of this gradient to 0, we have:

$$\mathrm{Diag}(\widehat{S}\mathrm{diag}(\alpha_r)B\mathrm{diag}(\alpha_c \odot \alpha_c)B^\top) = \mathrm{Diag}(SW\mathrm{diag}(\alpha_c)B^\top). \tag{28}$$

Let $C = B\mathrm{diag}(\alpha_c \odot \alpha_c)B^\top$, then we have:

$$\begin{aligned}
&\mathrm{Diag}(\widehat{S}\mathrm{diag}(\alpha_r)C) = \mathrm{Diag}(SW\mathrm{diag}(\alpha_c)B^\top) \\
\implies &(\widehat{S} \odot C)\alpha_r = \mathrm{Diag}(SW\mathrm{diag}(\alpha_c)B^\top) \\
\implies &\alpha_r = \big(\widehat{S} \odot C\big)^{-1} \mathrm{Diag}\big(SW\mathrm{diag}(\alpha_c)B^\top\big)
\end{aligned} \tag{29}$$

Derivation for B Explanding the objective loss $\mathcal{L}(.)$ out we have:

$$\begin{aligned}
&\mathrm{Tr}\left[(XW - \widehat{X}\widehat{W})(XW - \widehat{X}\widehat{W})^T\right] \\
&= \mathrm{Tr}\left[\widehat{X}\widehat{W}\widehat{W}^\top\widehat{X}^\top - 2XW\widehat{W}^\top\widehat{X}^\top + \mathrm{const}\right] \\
&\propto \mathrm{Tr}\left[\widehat{X}^\top\widehat{X}\widehat{W}\widehat{W}^\top\right] - 2\,\mathrm{Tr}\left[\widehat{X}^\top XW\widehat{W}^\top\right] \\
&= \mathrm{Tr}\left[\widehat{S}\widehat{W}\widehat{W}^\top\right] - 2\,\mathrm{Tr}\left[SW\widehat{W}^\top\right] \\
&= \mathrm{Tr}\left[\mathrm{diag}(\alpha_r)\widehat{S}\mathrm{diag}(\alpha_r)B\mathrm{diag}(\alpha_c \odot \alpha_c)B^\top\right] - 2\,\mathrm{Tr}\left[\mathrm{diag}(\alpha_r)SW\mathrm{diag}(\alpha_c)B^\top\right] \\
&= \mathrm{Tr}\left[NBKB^\top\right] - 2\,\mathrm{Tr}\left[PB^\top\right]
\end{aligned} \tag{30}$$

Table 5: Quantization time comparison between 1-bit LLM methods and ours across different models.

Model	ARB-X	ARB-RC	Ours
OPT-6.7B	87m	65m	93m
Llama-2-7B	91m	54m	76m
Llama-2-13B	147m	100m	119m

Table 6: Inference throughput (tokens/s) on the Intel Xeon Gold 6548Y+ CPU Wang et al. (2023). Higher is better.

Model	Prefill	Decode
FP16	2.83	0.87
ARB-RC	11.44	14.15
Ours	12.42	14.52

with $N = \text{diag}(\alpha_r)\widehat{S}\text{diag}(\alpha_r)$, $K = \text{diag}(\alpha_c \odot \alpha_c)$, $P = \text{diag}(\alpha_r)SW\text{diag}(\alpha_c)$.

We denote $\widetilde{B}_{-i}$ and $\widetilde{N}_{-i}$ respectively as the matrix B and N exclude the row i. Similarly, $\widetilde{N}_{i,-j}$ denotes the row i of matrix N excluded the j element. Given a row i of B, keeping the other row of B as constant, the loss can be expanded as a function of $B_{i,:}$ as:

$$\begin{aligned}\mathcal{L}_{MSE}(X, L) &\propto \text{Tr}\left[NBKB^\top\right] - 2\,\text{Tr}\left[PB^\top\right] \\ &= \sum_j \left(N_{j,:}BKB_{j,:}^\top - 2P_{j,:}B_{j,:}^\top\right) \\ &\propto N_{i,i}B_{i,:}KB_{i,:}^\top + 2\widetilde{N}_{i,-i}\widetilde{B}_{-i}KB_{i,:}^\top - 2P_{i,:}B_{i,:}^\top \\ &\propto \left(\widetilde{N}_{i,-i}\widetilde{B}_{-i}K - P_{i,:}\right)B_{i,:}^\top,\end{aligned} \tag{31}$$

Since $B_{i,:}$ is a binary vector, it has a closed-form solution:

$$B_{i,:} = \text{sign}\left(P_{i,:} - \widetilde{N}_{i,-i}\widetilde{B}_{-i}K\right). \tag{32}$$

The more compacted form of the whole B using this closed-form for each row can be computed as:

$$B = \text{sign}(P - [N - \text{diag}(\text{Diag}(N))]BK). \tag{33}$$

E Additional results

Quantization overhead. We report the quantization time of our method compared with ARB-X and ARB-RC Li et al. (2025b) across different architectures. Our method incurs additional offline quantization cost relative to ARB-RC, while remaining comparable to ARB-X, it is faster than ARB-X on both LLaMA models and slightly slower on OPT-6.7B. In terms of memory usage, for the Llama-2-7B model, our method requires a peak GPU memory of 12.1 GB, compared with 11.4 GB for ARB-X and 10.8 GB for ARB-RC. As quantization is performed only once offline, this modest computational and memory overhead is modest in practice.

E.1 Additional results on the efficiency of the model

We provide the throughput improvements of the Prefill and Decode passes (single batch with 2048 tokens) for quantized models, relative to their FP16 counterparts. The base model used is LLaMA-2-7B, with inferences conducted on a Intel Xeon Gold 6548Y+ CPU. Overall, our method achieves at least 4.4x speed up in both decoding and prefill compared to the FP counterpart, and comparable to the ARB-RC.

Table 7: Ablation study on the calibration set size for Qwen-2-1.5B.

Size	Method	PPL (↓)	
		C4	WikiText2
32	ARB-RC	109.45	95.08
	Ours	**83.44**	**76.56**
128	ARB-RC	109.83	89.59
	Ours	**74.83**	**68.45**
512	ARB-RC	127.84	89.13
	Ours	**71.27**	**54.66**
1024	ARB-RC	117.49	92.29
	Ours	**67.80**	**60.49**

E.2 Additional implementation details

It is widely known that block-level loss are more important than layer-wise loss, due to cross-layer dependency across layers within blocks (Li et al., 2021). However, naïvely applying output-driven quantization to all layers within a block does not necessarily guarantee minimization of the block-level loss. To ensure stability, we therefore adopt a conservative design choice and restrict output alignment to only the last fully connected layer of each block, which has a direct and dominant influence on the block loss. The remaining layers within each block are quantized using weight-alignment objective of ARB-RC Li et al. (2025b).

Ablation study for the size of the calibration set. We provide additional ablation study about the effect of our method when the size of the calibration set varies. The results are provided in Table 7. The results demonstrate that our method are effective both when the number of calibration samples is small, or when we use more data.

Ablation study for the value of the β. We provide additional ablation study about the value of β. The results are provided in Table 8. In practice we set the value of $\beta = 1$ for simplicity. We use a single fixed seed for this ablation study.

Table 8: Ablation study for the hyper-parameter β.

Model	β	PPL (↓)	
		C4	WikiText2
OPT-6.7B	0	16.24	14.56
	0.5	16.21	14.53
	1	16.18	14.59

F Proposed learning algorithm

Algorithm 1 Our-RC with Second-Order Relational Projection

procedure OUR-RC($W, S, \widehat{S}, T, k, \beta$)
  ▷ $W \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$: *full-precision weight* ◁
  ▷ $S = \widehat{X}^\top X \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}$ ◁
  ▷ $\widehat{S} = \widehat{X}^\top \widehat{X} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}$ ◁
  ▷ T: *refinement rounds;* k: *column-scale update interval* ◁
  ▷ β: *diagonal curvature multiplier* ◁
  Initialize α_r, α_c, B
  $\widehat{W} \leftarrow \mathrm{Diag}(\alpha_r) B \,\mathrm{Diag}(\alpha_c)$
  $M_G \leftarrow S W W^\top S^\top$
  for iter $= 1, \ldots, T$ **do**
    $\alpha_r^\star \leftarrow$ REFINE-$\alpha_r(S, \widehat{S}, W, B, \alpha_c)$
    $\mathcal{M}^r \leftarrow$ SO-$\alpha_r(M_G, B, \alpha_r, \alpha_c, \alpha_r^\star, \beta)$
    $\widetilde{\alpha}_r \leftarrow (1 - \mathcal{M}^r) \odot \alpha_r + \mathcal{M}^r \odot \alpha_r^\star$
    $\widehat{W}_{\text{new}} \leftarrow \mathrm{Diag}(\widetilde{\alpha}_r) B \,\mathrm{Diag}(\alpha_c)$
    if ACCEPT($\widehat{W}, \widehat{W}_{\text{new}}, W, S, \widehat{S}, M_G$) **then**
      $\alpha_r \leftarrow \widetilde{\alpha}_r$
      $\widehat{W} \leftarrow \widehat{W}_{\text{new}}$
    if $(\text{iter} + 1) \bmod k = 0$ **then**
      $\alpha_c^\star \leftarrow$ REFINE-$\alpha_c(S, \widehat{S}, W, B, \alpha_r)$
      $\mathcal{M}^c \leftarrow$ SO-$\alpha_c(M_G, B, \alpha_r, \alpha_c, \alpha_c^\star, \beta)$
      $\widetilde{\alpha}_c \leftarrow (1 - \mathcal{M}^c) \odot \alpha_c + \mathcal{M}^c \odot \alpha_c^\star$
      $\widehat{W}_{\text{new}} \leftarrow \mathrm{Diag}(\alpha_r) B \,\mathrm{Diag}(\widetilde{\alpha}_c)$
      if ACCEPT($\widehat{W}, \widehat{W}_{\text{new}}, W, S, \widehat{S}, M_G$) **then**
        $\alpha_c \leftarrow \widetilde{\alpha}_c$
        $\widehat{W} \leftarrow \widehat{W}_{\text{new}}$
    for $q = 1, 2$ **do**
      $(B^\star, i) \leftarrow$ REFINE-B($S, \widehat{S}, W, B, \alpha_r, \alpha_c$)
      $\mathcal{M}^B_{i,:} \leftarrow$ SO-B($M_G, B, B^\star, i, \alpha_r, \alpha_c, \beta$)
      $\widetilde{B} \leftarrow B$
      $\widetilde{B}_{i,:} \leftarrow (1 - \mathcal{M}^B_{i,:}) \odot B_{i,:} + \mathcal{M}^B_{i,:} \odot B^\star_{i,:}$
      $\widehat{W}_{\text{new}} \leftarrow \mathrm{Diag}(\alpha_r) \widetilde{B} \,\mathrm{Diag}(\alpha_c)$
      if ACCEPT($\widehat{W}, \widehat{W}_{\text{new}}, W, S, \widehat{S}, M_G$) **then**
        $B \leftarrow \widetilde{B}$
        $\widehat{W} \leftarrow \widehat{W}_{\text{new}}$
  $\widehat{W} \leftarrow \mathrm{Diag}(\alpha_r) B \,\mathrm{Diag}(\alpha_c)$
  return $\widehat{W}$

Algorithm 2 Auxiliary Functions

procedure REFINE-$\alpha_r(S, \widehat{S}, W, B, \alpha_c)$
  $K \leftarrow \mathrm{Diag}(\alpha_c \odot \alpha_c)$
  $R \leftarrow BKB^\top$
  $H_r^Q \leftarrow \widehat{S} \odot R$
  $b_r^Q \leftarrow \mathrm{diag}\left(SW\,\mathrm{Diag}(\alpha_c)B^\top\right)$
  $\alpha_r^\star \leftarrow \textsc{Solve}(H_r^Q, b_r^Q)$
  return $\alpha_r^\star$

procedure REFINE-$\alpha_c(S, \widehat{S}, W, B, \alpha_r)$
  $D_r \leftarrow \mathrm{Diag}(\alpha_r)$
  $\mathrm{num} \leftarrow \mathrm{diag}\left(B^\top D_r SW\right)$
  $\mathrm{den} \leftarrow \mathrm{diag}\left(B^\top D_r \widehat{S} D_r B\right)$
  $\alpha_c^\star \leftarrow \mathrm{num}/(\mathrm{den} + \varepsilon)$
  return $\alpha_c^\star$

procedure REFINE-B$(S, \widehat{S}, W, B, \alpha_r, \alpha_c)$
  $D_r \leftarrow \mathrm{Diag}(\alpha_r)$
  $D_c \leftarrow \mathrm{Diag}(\alpha_c)$
  $N \leftarrow D_r \widehat{S} D_r$
  $K \leftarrow D_c^2$
  $P \leftarrow D_r SW D_c$
  $N_{\text{off}} \leftarrow N - \mathrm{Diag}(\mathrm{diag}(N))$
  $Z \leftarrow P - N_{\text{off}} BK$
  $B^\star \leftarrow \mathrm{sign}(Z)$
  $\mathrm{gain}_j \leftarrow \sum_\ell \left(|Z_{j\ell}| - B_{j\ell} Z_{j\ell}\right)$
  $i \leftarrow \mathrm{argmax}_j\, \mathrm{gain}_j$
  return $B^\star, i$

procedure SO-$\alpha_r(M_G, B, \alpha_r, \alpha_c, \alpha_r^\star, \beta)$
  $D_c \leftarrow \mathrm{Diag}(\alpha_c)$
  $\widehat{W} \leftarrow \mathrm{Diag}(\alpha_r) B D_c$
  $\Delta_r \leftarrow \alpha_r^\star - \alpha_r$
  $U \leftarrow BD_c$
  $g_r \leftarrow \mathrm{diag}\left(M_G \widehat{W} D_c B^\top\right)$
  $h_r \leftarrow \mathrm{diag}(M_G) \odot \mathrm{diag}(UU^\top)$
  $\mathcal{M}^r \leftarrow \mathbf{1}\left[g_r \odot \Delta_r + \frac{\beta}{2} h_r \odot \Delta_r^{\odot 2} \geq 0\right]$
  return $\mathcal{M}^r$

procedure SO-$\alpha_c(M_G, B, \alpha_r, \alpha_c, \alpha_c^\star, \beta)$
  $D_r \leftarrow \mathrm{Diag}(\alpha_r)$
  $\widehat{W} \leftarrow D_r B\,\mathrm{Diag}(\alpha_c)$
  $\Delta_c \leftarrow \alpha_c^\star - \alpha_c$
  $V \leftarrow D_r B$
  $g_c \leftarrow \mathrm{diag}\left(B^\top D_r M_G \widehat{W}\right)$
  $h_c \leftarrow \mathrm{diag}\left(V^\top M_G V\right)$
  $\mathcal{M}^c \leftarrow \mathbf{1}\left[g_c \odot \Delta_c + \frac{\beta}{2} h_c \odot \Delta_c^{\odot 2} \geq 0\right]$
  return $\mathcal{M}^c$

procedure SO-B$(M_G, B, B^\star, i, \alpha_r, \alpha_c, \beta)$
  $D_r \leftarrow \mathrm{Diag}(\alpha_r)$
  $D_c \leftarrow \mathrm{Diag}(\alpha_c)$
  $\widehat{W} \leftarrow D_r B D_c$
  $\Delta_B \leftarrow B^\star_{i,:} - B_{i,:}$
  $g_B \leftarrow \left(D_r M_G \widehat{W} D_c\right)_{i,:}$
  $h_B \leftarrow \alpha_{r,i}^2 (M_G)_{ii} (\alpha_c \odot \alpha_c)$
  $\mathcal{M}^B_{i,:} \leftarrow \mathbf{1}\left[g_B \odot \Delta_B + \frac{\beta}{2} h_B \odot \Delta_B^{\odot 2} \geq 0\right]$
  return $\mathcal{M}^B_{i,:}$

procedure ACCEPT$(\widehat{W}_{\text{old}}, \widehat{W}_{\text{new}}, W, S, \widehat{S}, M_G)$
  $\mathcal{L}_{\text{old}} \leftarrow \mathrm{Tr}(\widehat{W}_{\text{old}}^\top \widehat{S} \widehat{W}_{\text{old}}) - 2\,\mathrm{Tr}(\widehat{W}_{\text{old}}^\top SW)$
  $\mathcal{L}_{\text{new}} \leftarrow \mathrm{Tr}(\widehat{W}_{\text{new}}^\top \widehat{S} \widehat{W}_{\text{new}}) - 2\,\mathrm{Tr}(\widehat{W}_{\text{new}}^\top SW)$
  $\mathcal{A}_{\text{old}} \leftarrow \frac{1}{2}\,\mathrm{Tr}(\widehat{W}_{\text{old}}^\top M_G \widehat{W}_{\text{old}})$
  $\mathcal{A}_{\text{new}} \leftarrow \frac{1}{2}\,\mathrm{Tr}(\widehat{W}_{\text{new}}^\top M_G \widehat{W}_{\text{new}})$
  return $\mathbf{1}[\mathcal{L}_{\text{new}} \leq \mathcal{L}_{\text{old}} \wedge \mathcal{A}_{\text{new}} \geq \mathcal{A}_{\text{old}}]$