# PATCH: PROPAGATION-AWARE TWO-STAGE QUANTIZATION ERROR COMPENSATION VIA FINE-GRAINED CORRECTION FOR HIGH-FIDELITY 2-BIT LLMS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Extreme low-bit quantization reduces the memory and data-movement costs of large language models (LLMs), but 2-bit precision introduces substantial errors that accumulate across layers. Existing compensation methods are constrained by both correction expressiveness and optimization scope: limited-rank corrections struggle to capture the error structure, channel-wise scaling overlooks intra-channel heterogeneity, and locally optimal compensation strengths need not minimize end-to-end loss. We introduce *PATCH*, a propagation-aware two-stage framework that decouples fine-grained correction from layer-wise strength regulation. The first stage fits group-wise compensation using activations from the quantized network, capturing heterogeneous intra-channel errors without an explicit rank constraint. The second stage fixes the learned corrections and jointly optimizes layer-wise regulation coefficients against an end-to-end objective, accounting for error propagation while preserving the correction directions. The resulting coefficients are fused into existing quantization parameters, avoiding a separate compensation branch. Across Gemma-2, Llama-3, and Qwen3 models of varying scales under 2-bit GPTQ, *PATCH* reduces WikiText-2 perplexity by 24.7% on average and improves average zero-shot accuracy by 2.03 percentage points relative to the best competing compensation method for each model and metric. These results demonstrate that jointly addressing correction expressiveness and network-wide compensation strength enables effective accuracy recovery in extreme low-bit LLMs.

## 1 INTRODUCTION

Recent research has focused on the efficient deployment of large language models (LLMs), which achieve high performance through extensive parameterization (Grattafiori et al., 2024; Yang et al., 2025). Among these approaches, quantization reduces the memory footprint of LLMs through low-bit representations (Kim et al., 2024; Gholami et al., 2022). In particular, growing use of personalized LLM applications has driven interest in on-device inference and research on extreme low-bit quantization (*i.e.,* 2-bit) for deploying LLMs on resource-constrained devices (Chee et al., 2023; Du et al., 2024). However, at extreme low-bit precision, quantization errors span multiple singular directions, vary within input channels, accumulate across layers, and severely degrade accuracy (Wang et al., 2026). Accordingly, various approaches have been explored to mitigate this trade-off between bit precision and accuracy (Frantar et al., 2022; Kang & Kim, 2025; Lin et al., 2024).

Quantization-error compensation mitigates accuracy degradation with minimal parameter overhead, achieving higher accuracy than conventional post-training quantization (PTQ) while incurring substantially lower optimization costs than full-model fine-tuning. LQER (Zhang et al., 2024), QERA (Zhang et al., 2025), SERQ (Park et al., 2026), and GlowQ (An et al., 2026) achieve this through low-rank corrections. Meanwhile, QwT (Fu et al., 2025) computes a linear compensation term through closed-form regression, whereas QwT-v2 (Tang et al., 2025) reduces the associated inference overhead by replacing it with vector compensation based on channel-wise scales and biases.

These methods have demonstrated that substantial accuracy recovery can be achieved with limited compensation cost, thereby mitigating performance degradation. However, under extreme low-bit



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.

quantization, the magnitude and structural complexity of quantization perturbations increase substantially, leaving the sufficiency of existing compensation methods uncertain. We identify two fundamental limitations of existing compensation methods by analyzing 2-bit quantization-error structure and the behavior of local and end-to-end objectives under varying compensation strengths that control the extent to which the optimized compensation is applied:

**Limited expressiveness of existing compensation methods.** The broad singular-value spectrum of 2-bit quantization errors challenges low-rank corrections, while channel-wise vector compensation cannot capture intra-channel heterogeneity because it shares one coefficient across groups within each channel. Thus, extreme low-bit quantization requires fine-grained, rank-unconstrained corrections that capture group-level errors (***Limitation 1***); see Section 3.1 for details.

**Mismatch between local and end-to-end objectives.** Existing compensation methods independently optimize each layer's local objective. However, in a quantized network, accumulated activation errors propagate across layers and interact, so a local optimum is not guaranteed to be optimal for the end-to-end objective. We find that the local and end-to-end objectives prefer different compensation strengths even without propagation error, while propagation error makes the end-to-end objective favor different compensation strengths across layers. Thus, compensation strength must be regulated per layer according to the end-to-end objective (***Limitation 2***); see Section 3.2 for details.

To address these limitations, we propose *PATCH*, a two-stage quantization-error compensation method that separately optimizes compensation expressiveness and strength. *PATCH* first optimizes fine-grained compensation under a local objective to capture the detailed structure of 2-bit quantization errors, and then regulates its layer-wise compensation strength according to an end-to-end objective while preserving the optimized compensation direction. This design captures quantization-error structure and inter-layer error propagation with minimal inference overhead, as the compensation coefficients are fused into the quantization parameters. Our contributions are as follows:

- **Fine-grained group-wise error compensation.** We introduce a fine-grained vector compensation scheme that assigns independent coefficients to each channel–group pair, directly correcting group-wise quantization errors within each channel. This design captures fine-grained errors in the 2-bit regime without imposing an explicit rank constraint or sharing a single coefficient across groups within a channel (Solution to ***Limitation 1***).
- **Propagation-aware compensation strength regulation.** We propose a propagation-aware regulation method that adjusts the compensation strength of each linear layer using an end-to-end objective to account for propagation error and the mismatch between local and end-to-end objectives. This approach determines layer-wise compensation strengths that minimize the end-to-end loss (Solution to ***Limitation 2***).
- **Practical applicability across PTQ methods and devices.** *PATCH* is compatible with diverse PTQ methods, including GPTQ, AWQ, and QuIP, and improves average zero-shot accuracy by 6.90 percentage points across six GPTQ-quantized models from the Gemma-2, Llama-3, and Qwen3 families. By fusing the compensation coefficients into the existing quantization parameters, *PATCH* eliminates the need for a separate compensation branch and incurs only 1% and 9% latency overhead relative to the GPTQ baseline on Gemma-2-2B when deployed on the B200 GPU and the Hexagon NPU, respectively.

## 2 Background

### 2.1 Quantization for LLMs

Quantization represents LLM weights at low-bit precision, reducing memory usage and data-movement costs. In particular, PTQ efficiently compresses pretrained models without retraining. The round-to-nearest (RTN) method rounds each weight to the nearest value on a predefined quantization grid. However, because it does not account for the impact of rounding errors on layer outputs or activation distributions, it can cause substantial accuracy degradation (Gholami et al., 2022).

To mitigate these limitations, GPTQ (Frantar et al., 2022) uses second-order information for sequential column-wise quantization and error compensation, AWQ (Lin et al., 2024) applies activation-aware channel-wise scaling, and QuIP (Chee et al., 2023) performs adaptive rounding after orthogonally transforming the weights and Hessian.

Despite these efforts, substantial quantization errors can persist under extreme low-bit precision due to its limited representational capacity. To address this limitation, recent methods recover lost information through lightweight compensation with minimal overhead.

### 2.2 Quantization Error Compensation Methods

**Low-rank error reconstruction.** Low-rank adapter-based reconstruction has emerged as a prominent approach to quantization-error compensation. It approximates the difference between the full-precision weight $\mathbf{W}$ and its quantized counterpart $\mathbf{W}_q$ with a low-rank component, reconstructing $\mathbf{W} \approx \mathbf{W}_q + \mathbf{AB}$ to recover accuracy with minimal inference overhead. Specifically, LQER (Zhang et al., 2024) facilitates low-rank error reconstruction through activation-aware scaling, while QERA (Zhang et al., 2025) computes compensation matrices from second-order activation statistics. SERQ (Park et al., 2026) targets errors according to activation and weight saliency, whereas QwT (Fu et al., 2025) applies matrix-and-bias compensation obtained via calibration-based least-squares regression. However, low-rank methods may lack capacity for extreme low-bit errors, while matrix-based compensation incurs additional inference cost.

**Vector Compensation.** QwT-v2 (Tang et al., 2025) reduces matrix-based compensation to channel-wise vector compensation using scales and biases. It applies an affine transformation to each quantized-output channel, with the compensation vectors obtained in closed form via one-dimensional regression. These vectors can be fused into the existing scaling and bias operations, minimizing inference overhead. Although this design recovers accuracy at low cost, compensation structures differ in expressiveness and optimization scope.

## 3 Motivation

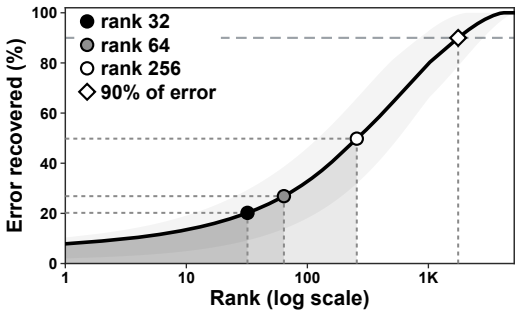
### 3.1 Limitations of Quantization-Error Representation

Under extreme 2-bit quantization, we first analyzed whether the resulting errors can be effectively represented with limited rank by applying GPTQ-based weight quantization to seven LLMs spanning different scales in the Llama and Qwen families: Llama-3.2-1B (Grattafiori et al., 2024), Llama-3.1-8B (Grattafiori et al., 2024), and Qwen3-1.7B/4B/8B/14B/32B (Yang et al., 2025). Using a calibration dataset, we estimated the second-moment matrix $\mathbf{C}_X = \mathbb{E}[\mathbf{x}\mathbf{x}^\top]$ and constructed the error matrix $\mathbf{E} = (\mathbf{W} - \mathbf{W}_q)\mathbf{C}_X^{1/2}$. We quantified the cumulative error energy captured by the first $r$ singular components as $\rho(r) = (\sum_{i=1}^{r} \sigma_i^2)/(\sum_i \sigma_i^2)$, where $\sigma_i$ are the singular values of $\mathbf{E}$. Figure 1a shows the cumulative error energy across ranks, with the solid line and shaded region denoting the median and distribution across seven models, respectively. Capturing 90% of the 2-bit quantization error requires approximately 1K ranks, making rank-$r$ compensation with $\mathrm{rank}(\mathbf{AB}) \leq r$ parameter-intensive. These findings highlight the need for correction structures that do not impose explicit rank constraints in extreme low-bit settings. Additional analysis is provided in Appendix C.1.

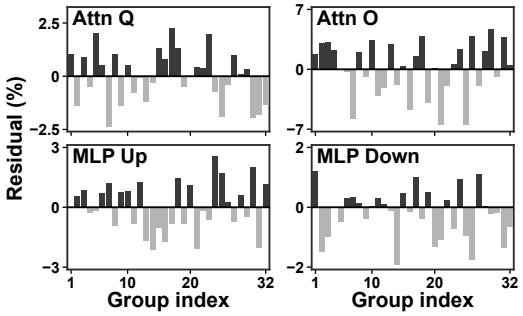
Meanwhile, vector compensation (Tang et al., 2025) applies a single compensation coefficient $M_c$ to each output channel $c$, yielding the error correction $\Delta\mathbf{W} = (\mathbf{M} - \mathbf{1}) \odot \mathbf{W}_q$. Although this approach does not impose a rank constraint, all quantization groups within a channel share the same coefficient. Figure 1b shows the residual distribution across quantization groups after channel-wise compensation. We measure each residual by projecting the post-compensation output error $r_{c,t}$ onto the corresponding group output direction $z_{c,g,t}$; its sign indicates whether the group is under- or over-corrected. We find that residual magnitudes and signs vary across groups, indicating that a single channel-wise coefficient cannot adequately compensate for intra-channel errors. This motivates fine-grained compensation with independent group-wise coefficients (***Limitation 1***).

### 3.2 Local-to-Global Objective Mismatch under Error Propagation

Existing compensation methods minimize the local weight-reconstruction error using clean activations (*i.e.*, BF16-network activations). However, local reconstruction quality does not necessarily translate into end-to-end performance. Specifically, the output discrepancy can be decomposed as follows, where $\tilde{\mathbf{x}}$ denotes the quantized-network activation containing propagated errors:

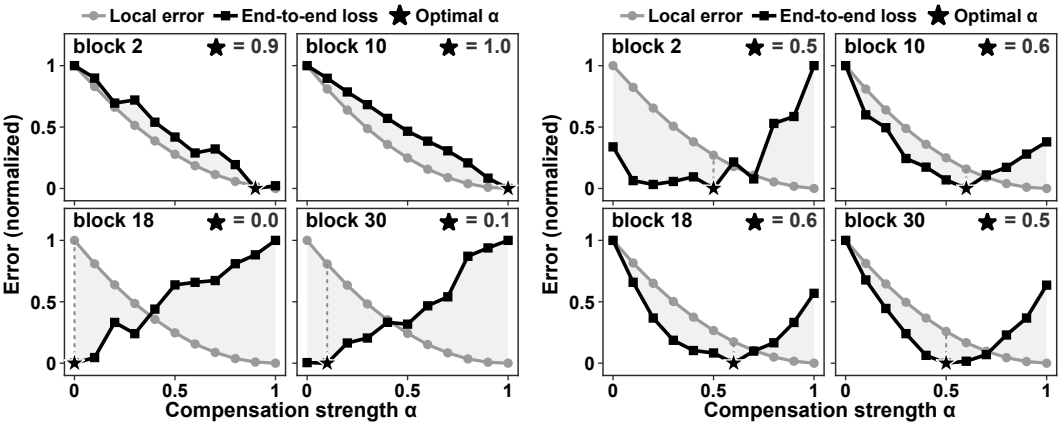



(a) Error reconstruction ratio with respect to compensation rank under 2-bit quantization.




(b) Intra-channel distribution of quantization-error residuals. ■ (black): under-corrected, the group needs a larger coefficient; ■ (gray): over-corrected, it needs a smaller one.

Figure 1: Characteristics of quantization error under 2-bit quantization.




(a) Fully quantized setting

(b) Isolated setting

Figure 2: Local and global loss over compensation strength $\alpha_\ell$ across transformer blocks.

$$\mathbf{W}\mathbf{x} - \mathbf{W}_q\tilde{\mathbf{x}} = (\mathbf{W} - \mathbf{W}_q)\tilde{\mathbf{x}} + \mathbf{W}(\mathbf{x} - \tilde{\mathbf{x}}). \tag{1}$$

The first term denotes local error, while the second captures propagation error from preceding layers. Thus, minimizing the local output error on clean activations does not necessarily minimize the final output error after subsequent layers. To test whether locally optimized compensation is also optimal end-to-end, we fixed the fine-grained compensation $\mathbf{M}_\ell$ for each linear layer $\ell$ and varied its application strength $\alpha_\ell$, measuring the resulting local error and end-to-end loss. The definition of $\mathbf{M}_\ell$ is provided in Section 4.1; here, we focus on varying its strength independently of its structure:

$$\mathbf{M}_\ell^{(\alpha_\ell)} = \mathbf{1} + \alpha_\ell \left(\mathbf{M}_\ell - \mathbf{1}\right), \qquad \alpha_\ell \in [0, 1]. \tag{2}$$

Figure 2 compares local error and end-to-end loss as functions of $\alpha_\ell$ for `q_proj` layers across different transformer blocks in Llama-3.1-8B. We further considered fully quantized and isolated settings to examine the effect of error propagation. In the isolated setting, only the target layer is quantized, while all other layers remain in full precision, preventing error propagation from preceding layers. In the fully quantized setting, all layers are quantized, with non-target compensation strengths fixed at 1. In both settings, local error decreased as $\alpha_\ell$ increased, whereas the end-to-end loss exhibited block-dependent optima that generally did not coincide with $\alpha_\ell = 1$. In particular, error propagation and inter-layer interactions caused the optimal strength to vary across blocks in the fully quantized setting. These results show that local objectives alone are insufficient and that compensation strength should be regulated per layer according to the end-to-end objective (***Limitation 2***).

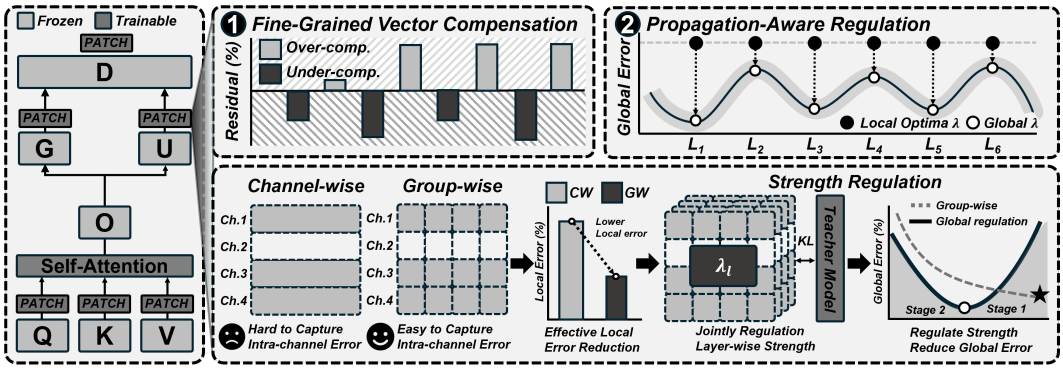



Figure 3: Overview of *PATCH*. Stage 1 sequentially optimizes fine-grained compensation using quantized-network activations, while Stage 2 freezes the optimized compensation and jointly optimizes layer-wise compensation-regulation coefficients with an end-to-end objective.

## 4 Proposed Methods

We propose *PATCH*, a two-stage compensation method for mitigating extreme low-bit quantization errors. As illustrated in Figure 3, it first applies fine-grained vector compensation to capture heterogeneous intra-channel errors without relying on a prescribed rank. The compensation coefficients are then fixed, while their application strengths are jointly optimized using an end-to-end objective. This design separates local error reconstruction from network-wide compensation-strength selection, thereby capturing both compensation expressiveness and inter-layer interactions.

### 4.1 Fine-grained Vector Compensation

We established the need for fine-grained vector compensation. Accordingly, we assign one compensation coefficient $m_{c,g}$ to each output channel $c$ and quantization group $g$. Specifically, denoting the quantization group containing input channel $i$ by $g(i)$, each element of the compensation matrix $\mathbf{M}$ is defined as follows:

$$M_{c,i} = m_{c,g(i)}, \qquad \hat{\mathbf{y}} = (\mathbf{M} \odot \mathbf{W}_q)\tilde{\mathbf{x}} + \boldsymbol{\beta}. \tag{3}$$

Here, $g(i)$ denotes the index of the quantization group containing input channel $i$, and $\boldsymbol{\beta}$ represents the output-channel bias correction. Accordingly, the correction $\Delta\mathbf{W} = (\mathbf{M} - \mathbf{1}) \odot \mathbf{W}_q$ is not constrained by a prescribed rank and provides finer-grained compensation than conventional channel-wise methods. To isolate activation-propagation discrepancies from local weight discrepancies and determine the weight-error correction for the actual quantized-network input, we optimize the following objective using activations $\tilde{\mathbf{x}}$ collected from the quantized network:

$$\min_{\{m_{c,g}\},\boldsymbol{\beta}} \mathbb{E}_{\tilde{\mathbf{x}}} \left[ \|(\mathbf{M} \odot \mathbf{W}_q)\tilde{\mathbf{x}} + \boldsymbol{\beta} - \mathbf{W}\tilde{\mathbf{x}}\|_2^2 \right]. \tag{4}$$

For a layer partitioned into $G$ groups, the trainable parameters consist of $\mathcal{O}(d_{\text{out}}G)$ group-wise coefficients and $\mathcal{O}(d_{\text{out}})$ bias terms. Layers are optimized sequentially with the compensated activations propagated to subsequent layers.

**Inference fusion and metadata overhead.** The optimized group-wise coefficient $m_{c,g(i)}$ is fused into the existing quantization scale, enabling its application without additional compensation operations. Specifically, dequantization of the quantized weight is modified as follows:

$$s^*_{c,g(i)} = m_{c,g(i)} s_{c,g(i)}, \qquad W^*_{q,c,i} = s^*_{c,g(i)}(q_{c,i} - z_{c,g(i)}). \tag{5}$$

Here, $q_{c,i}$ denotes the quantized integer weight, while $s_{c,g(i)}$ and $z_{c,g(i)}$ denote the scale and zero-point, respectively, of the quantization group $g(i)$ containing input channel $i$. The group-wise coefficient $m_{c,g(i)}$ is fused into the existing quantization scale and therefore requires no additional metadata. Likewise, $\boldsymbol{\beta}$ is fused into the existing linear-layer bias, while bias-free layers require only $\mathcal{O}(d_{\text{out}})$ additional bias terms. Consequently, compensation can be incorporated into dequantization without an additional correction matrix multiplication.

### 4.2 Propagation-aware Compensation Regulation

As shown in Section 3.2, local and end-to-end objectives may favor different compensation strengths across layers due to propagation errors. Accordingly, Stage 2 regulates the layer-wise strength with which compensation is applied while preserving its direction. For each linear layer $\ell$, we introduce a learnable scalar $\theta_\ell$ and define the regulation coefficient as $\lambda_\ell = \sigma(\theta_\ell) \in (0, 1)$, where $\sigma(\cdot)$ denotes the sigmoid function. The coefficient $\lambda_\ell$ interpolates between no compensation and the learned matrix $\mathbf{M}_\ell$, with the same regulation applied to the compensation bias:

$$\mathbf{M}_\ell^{(\lambda_\ell)} = \mathbf{1} + \lambda_\ell \left(\mathbf{M}_\ell - \mathbf{1}\right), \qquad \boldsymbol{\beta}_\ell^{(\lambda_\ell)} = \lambda_\ell \boldsymbol{\beta}_\ell. \tag{6}$$

Accordingly, the output of layer $\ell$ after regulation is given by:

$$\hat{\mathbf{y}}_\ell^{(\lambda_\ell)} = \left(\left[\mathbf{1} + \lambda_\ell \left(\mathbf{M}_\ell - \mathbf{1}\right)\right] \odot \mathbf{W}_{q,\ell}\right)\tilde{\mathbf{x}}_\ell + \lambda_\ell \boldsymbol{\beta}_\ell = \left(\mathbf{M}_\ell^{(\lambda_\ell)} \odot \mathbf{W}_{q,\ell}\right)\tilde{\mathbf{x}}_\ell + \boldsymbol{\beta}_\ell^{(\lambda_\ell)}. \tag{7}$$

Thus, $\lambda_\ell \to 0$ and $\lambda_\ell \to 1$ recover the quantized baseline and first-stage compensation, respectively, enabling layer-wise regulation of compensation strength while preserving its direction.

**Joint end-to-end optimization.** During optimization, we fix the quantized weights and the first-stage parameters $m_{c,g(i)}$ and $\boldsymbol{\beta}$, and jointly optimize all regulation parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N)$ for a model with $N$ linear layers. Let $p_t^{\mathrm{FP}}$ and $p_t(\boldsymbol{\theta})$ denote the output token distributions of the full-precision and regulated quantized models, respectively. We minimize the following KL divergence:

$$\min_{\boldsymbol{\theta}} \mathcal{L} = \frac{1}{T} \sum_{t=1}^{T} \mathrm{KL}\left(p_t^{\mathrm{FP}} \,\middle\|\, p_t(\boldsymbol{\theta})\right). \tag{8}$$

Here, $T$ denotes the number of valid token positions. Since the gradients of all regulation terms are computed through the entire quantized network, they reflect error propagation across layers. The optimized coefficients $\lambda_\ell^* = \sigma(\theta_\ell^*)$ are absorbed into the compensation scales and biases:

$$\mathbf{M}_\ell^* = \mathbf{1} + \lambda_\ell^* \left(\mathbf{M}_\ell - \mathbf{1}\right), \qquad \boldsymbol{\beta}_\ell^* = \lambda_\ell^* \boldsymbol{\beta}_\ell. \tag{9}$$

During inference, only the fixed $\mathbf{M}_\ell^*$ and $\boldsymbol{\beta}_\ell^*$ are used, requiring no additional inference operators. Thus, Stage 2 preserves the direction of fine-grained compensation while regulating its layer-wise strength according to the end-to-end objective. Through this two-stage optimization, *PATCH* captures intra-channel error heterogeneity via fine-grained correction and regulates the compensation strength layer by layer using an end-to-end objective, thereby recovering the accuracy of the quantized model with only the compensation bias as additional inference overhead.

## 5 Experimental Results

### 5.1 Environment Setting

We evaluated *PATCH* on Llama-3.2-1B (Grattafiori et al., 2024), Llama-3.1-8B (Grattafiori et al., 2024), Qwen3-8B/14B (Yang et al., 2025), and Gemma-2-2B/9B (Team et al., 2024). Using GPTQ (Frantar et al., 2022), AWQ (Lin et al., 2024), and QuIP (Chee et al., 2023) as base PTQ methods, we applied 2-bit weight-only quantization with group size 128 (W2G128). For QuIP, which lacks an explicit group-quantization configuration, compensation groups were defined over 128 input channels. Performance was measured by WikiText-2 (Merity et al., 2016) perplexity (PPL) and average accuracy over eight zero-shot tasks: BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC-Easy/Challenge (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018), and SIQA (Sap et al., 2019).

All quantization and compensation optimization was performed on an NVIDIA B200 GPU. We used the same calibration set for all methods, comprising 128 sequences of 2048 tokens sampled from the C4 training set (Raffel et al., 2020). In Stage 1, we computed the coefficients by solving Eq. 4 in closed form. For Stage 2, we initialized all $\lambda_\ell$ to $0.5$ and optimized the regulation parameters with Adam (Kingma & Ba, 2014) at a learning rate of $0.05$ for 60 steps, using four C4 samples that did not overlap with the calibration set. For all comparison methods, we used the official implementations with only the bit precision set to 2 bits. LQER/L$^2$QER differ in activation usage, whereas QERA-E/QERA-A differ in error-matrix construction. We set $r = 32$ for LQER, L$^2$QER, QERA-A, and QERA-E, and used the paper defaults for GlowQ ($r = 64$) and SERQ ($r = 128$). L$^2$QER and QERA-E were used for accuracy comparisons, while optimization costs were measured for all methods. Further experiments are provided in Appendices A and B.

Table 1: WikiText-2 PPL comparison of 2-bit quantization-error compensation methods on GPTQ.

| Model | BF16 | GPTQ | L$^2$QER | QERA-E | SERQ | GlowQ | QwT | QwT-v2 | PATCH |
|---|---|---|---|---|---|---|---|---|---|
| **Llama Family** | | | | | | | | | |
| Llama-3.2-1B | 12.90 | 6,171.7 | 6,487.6 | 4,497.8 | 3,362.8 | 4,266.4 | 3,498.3 | 1,499.3 | **582.7** |
| Llama-3.1-8B | 8.07 | 292.1 | 187.6 | 172.2 | 210.0 | 200.1 | 640.0 | 349.8 | **85.37** |
| **Qwen Family** | | | | | | | | | |
| Qwen3-8B | 10.98 | 59.34 | 47.55 | 48.20 | 44.30 | 47.55 | 34.37 | 76.08 | **28.37** |
| Qwen3-14B | 8.89 | 64.97 | 29.81 | 40.54 | 29.39 | 29.97 | **26.51** | 33.92 | 26.52 |
| **Gemma Family** | | | | | | | | | |
| Gemma-2-2B | 11.40 | 168.9 | 173.8 | 111.8 | 103.7 | 107.7 | 66.00 | 105.8 | **54.43** |
| Gemma-2-9B | 8.99 | 32.77 | 30.20 | 30.14 | 26.36 | 29.77 | 22.96 | 36.27 | **22.51** |

Table 2: WikiText-2 PPL comparison across different 2-bit PTQ methods.

| | | AWQ | | | | | | QuIP | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | BF16 | Base$^\dagger$ | L$^2$QER | QERA-E | GlowQ | QwT-v2 | PATCH | Base$^\dagger$ | L$^2$QER | QERA-E | GlowQ | QwT-v2 | PATCH |
| **Llama Family** | | | | | | | | | | | | | |
| Llama-3.2-1B | 12.90 | 2.06e5 | 1.49e5 | 4.00e4 | 4.10e4 | 9.36e3 | **1.77e3** | 2.23e3 | 3.01e3 | 2.56e3 | 2.16e3 | 792.1 | **281.6** |
| Llama-3.1-8B | 8.07 | 5.29e5 | 6.36e4 | 2.67e3 | 1.37e3 | 5.47e3 | **382.6** | 221.9 | 263.0 | 218.4 | 221.1 | 175.3 | **40.34** |
| **Qwen Family** | | | | | | | | | | | | | |
| Qwen3-8B | 10.98 | 6.57e4 | 920.5 | 375.0 | 121.0 | 532.7 | **41.24** | 3.23e3 | 88.25 | 60.46 | 51.97 | 52.49 | **28.59** |
| Qwen3-14B | 8.89 | 8.50e3 | 778.6 | 114.0 | 94.07 | 450.0 | **73.81** | 58.29 | 30.50 | 27.77 | 25.98 | 36.90 | **25.64** |
| **Gemma Family** | | | | | | | | | | | | | |
| Gemma-2-2B | 11.40 | 2.04e5 | 4.81e3 | 895.9 | 594.4 | 989.0 | **186.5** | 69.53 | 60.96 | 54.47 | 52.13 | 58.64 | **34.29** |
| Gemma-2-9B | 8.99 | 1.07e4 | 306.5 | 53.53 | 45.17 | 137.9 | **33.78** | 22.25 | 21.15 | 21.17 | 20.82 | 22.27 | **17.31** |

$^\dagger$ Base denotes the corresponding PTQ method without quantization-error compensation.

### 5.2 LLM Accuracy across PTQ Methods and Downstream Tasks

We first compared quantization-error compensation methods on WikiText-2 under identical quantization and calibration settings. As shown in Table 1, *PATCH* achieves lower PPL than the GPTQ baseline across all Llama, Qwen, and Gemma models and outperforms prior methods on most models. In particular, it attains PPLs of 582.7 and 85.37 on Llama-3.2-1B and Llama-3.1-8B, respectively, substantially improving over the previous state of the art, while achieving performance comparable to QwT on Qwen3-14B. Replacing GPTQ with AWQ and QuIP further verifies its compatibility: as shown in Table 2, *PATCH* effectively compensates for the residual 2-bit quantization errors and achieves the best performance under both PTQ methods. These results demonstrate robust generalization across model scales, architectures, and PTQ methods. Rank-sweep comparisons with low-rank methods are provided in Appendix C.2.

We next evaluated downstream performance by measuring average accuracy across eight zero-shot tasks on GPTQ-based models. As shown in Table 3, *PATCH* achieved the highest accuracy on every model, reaching 47.29%, a 6.90-percentage-point improvement over the GPTQ baseline. This recovers approximately 26.2% of the 26.29-percentage-point gap between BF16 and GPTQ, substantially mitigating the accuracy degradation caused by 2-bit quantization. This demonstrates effective recovery of downstream language-understanding performance at the 2-bit precision targeted for efficient on-device deployment.

### 5.3 Optimization Cost and Metadata Overhead Comparison

Table 4 compares the optimization time and additional deployment metadata required by each compensation method under 2-bit GPTQ. *PATCH* requires 1.5 and 2.8 minutes for Gemma-2-2B and Llama-3.1-8B, respectively, reducing optimization time by 76.9% and 83.3% relative to SERQ. Although it requires 0.5 and 1.6 additional minutes compared with QwT-v2, respectively, both optimization stages are completed within three minutes. Its deployment metadata is only 1.35 and 2.63 MiB, matching QwT-v2 and representing over 98% less storage than adapter-based methods, which require 79.22–263.36 MiB and 160.00–1024.25 MiB, respectively. By fusing group-wise compensation coefficients into existing quantization scales and adding only a few output-channel bias terms, *PATCH* enables fine-grained vector compensation with low optimization cost and minimal deployment overhead, facilitating practical deployment.

Table 3: Average zero-shot accuracy (%) across eight downstream tasks using different 2-bit quantization-error compensation methods on GPTQ.

| Model | BF16 | GPTQ | L$^2$QER | QERA-E | SERQ | GlowQ | QwT | QwT-v2 | PATCH |
|---|---|---|---|---|---|---|---|---|---|
| **Llama Family** | | | | | | | | | |
| Llama-3.2-1B | 55.28 | 35.71 | 35.64 | 35.63 | 35.46 | 35.32 | 35.97 | 36.54 | **38.10** |
| Llama-3.1-8B | 68.58 | 36.81 | 37.87 | 37.81 | 36.58 | 37.45 | 35.01 | 37.77 | **41.33** |
| **Qwen Family** | | | | | | | | | |
| Qwen3-8B | 68.36 | 39.46 | 41.13 | 40.69 | 40.64 | 41.15 | 44.10 | 47.56 | **49.97** |
| Qwen3-14B | 71.17 | 42.51 | 43.97 | 43.82 | 44.52 | 44.61 | 47.09 | 52.93 | **54.61** |
| **Gemma Family** | | | | | | | | | |
| Gemma-2-2B | 64.57 | 38.49 | 39.73 | 39.64 | 41.23 | 41.05 | 41.74 | 43.09 | **44.72** |
| Gemma-2-9B | 72.14 | 49.35 | 50.82 | 50.91 | 50.94 | 50.55 | 53.52 | 51.51 | **54.99** |
| **Average across Models** | | | | | | | | | |
| Average | 66.68 | 40.39 | 41.53 | 41.42 | 41.56 | 41.69 | 42.91 | 44.90 | **47.29** |

Table 4: Optimization time and deployment metadata under 2-bit GPTQ.

| | Gemma-2-2B | | Llama-3.1-8B | |
|---|---|---|---|---|
| Method | Time (min) | Meta. (MiB) | Time (min) | Meta. (MiB) |
| *Adapter-based Methods* | | | | |
| LQER | 0.59 | 79.22 | 2.9 | 160.00 |
| L$^2$QER | 0.69 | | 3.1 | |
| QERA-E | 2.0 | | 7.8 | |
| QERA-A | 0.70 | | 3.1 | |
| SERQ | 6.5 | 173.47 | 16.8 | 338.48 |
| GlowQ | 0.95 | 136.50 | 3.0 | 272.00 |
| QwT | 2.1 | 263.36 | 3.8 | 1024.25 |
| *Vector Compensation* | | | | |
| QwT-v2 | 1.0 | 1.35 | 1.2 | 2.63 |
| **PATCH** | **1.5** | **1.35** | **2.8** | **2.63** |

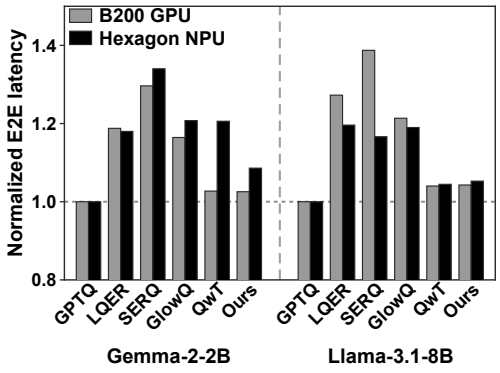



Figure 4: Normalized E2E latency on B200 GPU and Hexagon NPU.

### 5.4 Latency Overhead on Server GPU and On-Device NPU

To assess the deployment feasibility and inference overhead of *PATCH*, we measured the end-to-end latency of a workload with 128-token prefill and 128-token decode on an NVIDIA B200 GPU and the Hexagon V81 NPU of the Snapdragon 8 Elite Gen 5 SoC (Qualcomm, 2025). On the GPU, we used the same GPTQ W2G128 models and packed-weight kernel, whereas the NPU used the QNN HTP backend of Qualcomm AI Runtime (QAIRT). Figure 4 reports the end-to-end latency normalized to the GPTQ baseline for each model and execution environment.

For Gemma-2-2B and Llama-3.1-8B, *PATCH* incurred latency overheads relative to GPTQ of $1.01\times$ and $1.04\times$ on the B200 GPU, and $1.09\times$ and $1.05\times$ on the NPU, respectively. In contrast, the three low-rank methods (*i.e.*, LQER, SERQ, and GlowQ) incurred average latency overheads relative to GPTQ of $1.22\times$ and $1.29\times$ on the B200 GPU, and $1.24\times$ and $1.18\times$ on the NPU, respectively. Accordingly, *PATCH* reduced the average overhead of these methods from 22.3% to 1.2% and from 29.2% to 4.3% on the B200 GPU, and from 24.3% to 8.6% and from 18.4% to 5.3% on the NPU, for Gemma-2-2B and Llama-3.1-8B, respectively.

### 5.5 Ablation Study

#### 5.5.1 Ablation on Compensation and Regulation Granularity

To evaluate the two-stage design of *PATCH* and the effect of regulation granularity, we conducted ablation studies on Llama-3.1-8B, Qwen3-8B, and Gemma-2-9B, using QuIP as the base PTQ method. Stage 1 compares rank-32 low-rank, channel-wise, and group-wise compensation, while Stage 2 compares no regulation with global, block-, and layer-wise regulation. As shown in Table 5, Stage 1 group-wise compensation achieves lower PPL and higher accuracy than low-rank and channel-wise

Table 5: Ablation study of compensation structure and regulation granularity.

| Compensation Structure | Regulation Granularity | Llama-3.1-8B PPL ↓ | Llama-3.1-8B Acc. ↑ | Qwen3-8B PPL ↓ | Qwen3-8B Acc. ↑ | Gemma-2-9B PPL ↓ | Gemma-2-9B Acc. ↑ |
|---|---|---|---|---|---|---|---|
| **Reference** | | | | | | | |
| None | None | 221.9 | 37.35 | 3,234.3 | 36.90 | 22.25 | 55.15 |
| **Compensation Structure** | | | | | | | |
| Low-rank ($r = 32$) | Layer | 136.6 | 37.51 | 58.82 | 41.49 | 21.12 | 56.30 |
| Channel | Layer | 44.82 | 42.84 | 30.97 | 49.92 | 17.71 | 59.40 |
| **Group** | Layer | **40.34** | **44.74** | **28.59** | **50.21** | **17.31** | **59.81** |
| **Regulation Granularity** | | | | | | | |
| Group | None | 124.9 | 39.42 | 49.43 | 45.46 | 21.08 | 58.51 |
| Group | Global | 55.06 | 41.95 | 33.82 | 48.76 | 18.12 | 59.00 |
| Group | Block | 50.32 | 41.40 | 30.71 | 49.75 | 17.86 | 59.65 |
| Group | **Layer** | **40.34** | **44.74** | **28.59** | **50.21** | **17.31** | **59.81** |

compensation across all models, demonstrating that fine-grained coefficients effectively capture heterogeneous intra-channel errors.

However, Stage 1 alone remained suboptimal, whereas adding Stage 2 regulation yielded further improvements. This result shows that fine-grained compensation provides sufficient error expressiveness, while layer-wise strength regulation is necessary for end-to-end accuracy. Moreover, layer-wise regulation achieved the lowest PPL and the highest zero-shot accuracy across all three models, outperforming global and block-wise regulation. These results validate the effectiveness of independently regulating each linear layer to account for layer-dependent error propagation. Additional ablation results across a broader range of quantization configurations are provided in Appendix B.3.

### 5.5.2 Effect of Strength Regulation on End-to-End Loss

Figure 5 visualizes how Stage 2 regulates the compensation strength across four `o_proj` layers of Llama-3.1-8B. The Stage-1 strength (△) is fixed at $\lambda_\ell = 1$, corresponding to fully applying the locally optimized compensation. After Stage 2, the optimized strength (○) shifts toward the end-to-end optimum (⋆), rather than remaining at $\lambda_\ell = 1$. This behavior is consistent across blocks and demonstrates that end-to-end optimization selects layer-dependent compensation strengths to reduce the global loss. This result demonstrates that Stage 2 is essential for selecting compensation strengths that are optimal from the end-to-end perspective.

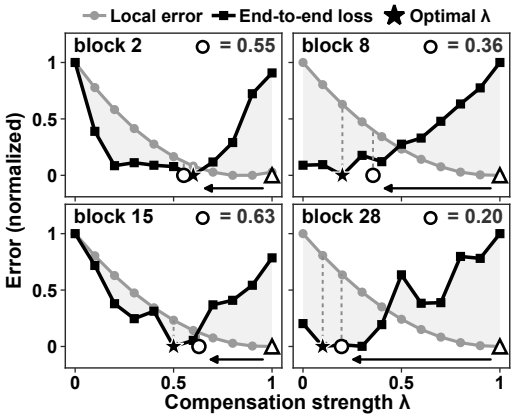



Figure 5: Stage-2 strength regulation on Llama-3.1-8B.

## 6 Conclusion

We presented *PATCH*, a two-stage method for achieving high accuracy and efficient deployment in 2-bit LLM quantization. It first corrects local weight errors through fine-grained group-wise compensation and then regulates the compensation strength of each layer using an end-to-end objective. Experiments on Llama, Qwen, and Gemma models show that *PATCH* achieves lower perplexity and higher downstream accuracy than prior methods across model scales and PTQ methods. By fusing the compensation coefficients into existing quantization parameters, *PATCH* requires no additional inference operators and incurs low overhead on both GPUs and on-device NPUs. These results demonstrate that jointly addressing fine-grained error compensation and propagation-aware strength regulation is effective for accurate and practical extreme low-bit LLM inference.

### AI USE STATEMENT

We used generative AI tools to assist with translation and to polish the writing of this paper, as well as to support the retrieval and discovery of potentially relevant literature. All AI-assisted text was reviewed and revised by the authors, and all retrieved references and related-work information were independently verified. The authors take full responsibility for the final content of this paper.

### REPRODUCIBILITY STATEMENT

The evaluated models, PTQ methods, quantization configurations, benchmarks, calibration data, baseline settings, and optimization hyperparameters are described in Section 5.1; the group-wise compensation objective, sequential optimization procedure, and inference-time fusion of compensation coefficients in Section 4.1; and the layer-wise regulation parameterization, end-to-end objective, and optimization procedure in Section 4.2. The hardware, runtime configurations, workloads, and procedures for measuring inference latency on the B200 GPU and Hexagon NPU are described in Section 5.4.

### ETHICS STATEMENT

This work studies quantization-error compensation for efficient low-bit LLM inference and involves no human-subject experiments or collection of new personal information. The experiments use existing pretrained models and established language-model benchmarks. The authors are responsible for the appropriate use of the resulting compressed models in downstream deployment settings.

## REFERENCES

Selim An, Il Suh, and Yeseong Kim. Glowq: Group-shared low-rank approximation for quantized llms. In *International Conference on Learning Representations*, volume 2026, pp. 124445–124486, 2026.

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.

Dayou Du, Yijia Zhang, Shijie Cao, Jiaqi Guo, Ting Cao, Xiaowen Chu, and Ningyi Xu. Bitdistiller: Unleashing the potential of sub-4-bit llms via self-distillation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 102–116, 2024.

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.

Minghao Fu, Hao Yu, Jie Shao, Junjie Zhou, Ke Zhu, and Jianxin Wu. Quantization without tears. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4462–4472. IEEE, 2025.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-power computer vision*, pp. 291–326. Chapman and Hall/CRC, 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.

Beom Jin Kang and Hyun Kim. Genptq: Green post-training quantization for large-scale asr models with mixed-precision bit allocation. In *EMNLP (Findings)*, pp. 10704–10718, 2025.

Nam Joon Kim, Jongho Lee, and Hyun Kim. Hyq: Hardware-friendly post-training quantization for cnn-transformer hybrid networks. In Kate Larson (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 4291–4299. International Joint Conferences on Artificial Intelligence Organization, 8 2024. doi: 10.24963/ijcai.2024/474. URL https://doi.org/10.24963/ijcai.2024/474. Main Track.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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.

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 *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

Yeonsik Park, Hyeonseong Kim, and Seungkyu Choi. Serq: Saliency-aware low-rank error reconstruction for llm quantization. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 35889–35909, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/3ca380c5fe9f174a71a230478741169f-Paper-Conference.pdf.

Qualcomm. Snapdragon 8 elite gen 5 mobile platform. https://www.qualcomm.com/smartphones/products/8-series/snapdragon-8-elite-gen-5, 2025. Accessed: Sep. 21, 2026.

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.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 4463–4473, 2019.

Ningyuan Tang, Minghao Fu, Hao Yu, and Jianxin Wu. Qwt-v2: Practical, effective and efficient post-training quantization. *arXiv preprint arXiv:2505.20932*, 2025.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Haodong Wang, Junjie Liu, Zicong Hong, Qianli Liu, Jian Lin, Song Guo, and Xu Chen. Twinquant: Learnable subspace decomposition for 4-bit llm quantization. *arXiv preprint arXiv:2606.01556*, 2026.

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.

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.

Cheng Zhang, Jianyi Cheng, George A Constantinides, and Yiren Zhao. Lqer: Low-rank quantization error reconstruction for llms. *arXiv preprint arXiv:2402.02446*, 2024.

Cheng Zhang, Jeffrey TH Wong, Can Xiao, George Constantinides, and Yiren Zhao. Qera: an analytical framework for quantization error reconstruction. In *International Conference on Learning Representations*, volume 2025, pp. 12531–12560, 2025.

Table 6: Average zero-shot accuracy (%) across eight downstream tasks using different 2-bit quantization-error compensation methods on AWQ.

| Model | BF16 | AWQ | L$^2$QER | QERA-E | GlowQ | QwT | QwT-v2 | PATCH |
|---|---|---|---|---|---|---|---|---|
| **Llama Family** | | | | | | | | |
| Llama-3.2-1B | 55.28 | 34.95 | **35.62** | 34.60 | 35.37 | 35.59 | 34.26 | 35.21 |
| Llama-3.1-8B | 68.58 | 35.39 | 35.03 | 35.41 | 35.77 | 35.96 | 34.37 | **37.15** |
| **Qwen Family** | | | | | | | | |
| Qwen3-8B | 68.36 | 35.32 | 36.36 | 38.58 | 39.43 | 43.92 | 40.64 | **48.23** |
| Qwen3-14B | 71.17 | 36.72 | 39.72 | 43.52 | 44.53 | 47.25 | 44.09 | **52.77** |
| **Gemma Family** | | | | | | | | |
| Gemma-2-2B | 64.57 | 35.05 | 36.04 | 38.39 | 40.02 | 40.61 | 38.49 | **41.23** |
| Gemma-2-9B | 72.14 | 39.13 | 42.82 | 46.82 | 48.04 | **51.79** | 41.62 | 51.28 |
| **Average across Models** | | | | | | | | |
| **Average** | **66.68** | **36.09** | **37.60** | **39.55** | **40.53** | **42.52** | **38.91** | **44.31** |

Table 7: Average zero-shot accuracy (%) across eight downstream tasks using different 2-bit quantization-error compensation methods on QuIP.

| Model | BF16 | QuIP | L$^2$QER | QERA-E | GlowQ | QwT | QwT-v2 | PATCH |
|---|---|---|---|---|---|---|---|---|
| **Llama Family** | | | | | | | | |
| Llama-3.2-1B | 55.28 | 35.48 | 35.62 | 35.32 | 35.20 | **35.75** | 34.91 | 35.62 |
| Llama-3.1-8B | 68.58 | 37.35 | 37.26 | 37.20 | 36.88 | 38.17 | 37.81 | **44.74** |
| **Qwen Family** | | | | | | | | |
| Qwen3-8B | 68.36 | 36.90 | 40.00 | 40.68 | 40.72 | 41.50 | 46.59 | **50.21** |
| Qwen3-14B | 71.17 | 43.71 | 47.37 | 48.64 | 48.64 | 50.05 | 56.76 | **58.34** |
| **Gemma Family** | | | | | | | | |
| Gemma-2-2B | 64.57 | 41.35 | 43.07 | 43.19 | 42.98 | 46.40 | 45.60 | **46.77** |
| Gemma-2-9B | 72.14 | 55.15 | 56.00 | 56.32 | 56.69 | 59.01 | 57.98 | **59.81** |
| **Average across Models** | | | | | | | | |
| **Average** | **66.68** | **41.66** | **43.22** | **43.56** | **43.52** | **45.15** | **46.61** | **49.25** |

# A Extended Main Results

## A.1 Extended Zero-shot Results

We further evaluate *PATCH* with AWQ and QuIP, reporting the average zero-shot accuracy across the same eight downstream tasks used in the main experiments. As shown in Tables 6 and 7, *PATCH* achieves the highest mean accuracy among the compared compensation methods under both PTQ baselines: 44.31% with AWQ and 49.25% with QuIP. These results improve over the corresponding uncompensated baselines by 8.22 and 7.59 percentage points, respectively. Under AWQ, *PATCH* achieves the best result on four of the six models, including gains of 4.31 points on Qwen3-8B and 5.52 points on Qwen3-14B over the strongest competing method for each model. Under QuIP, it leads on five models, with the largest gain of 6.57 points on Llama-3.1-8B. The exceptions are Llama-3.2-1B and Gemma-2-9B under AWQ, and Llama-3.2-1B under QuIP, where another compensation method performs slightly better. Overall, the results show that *PATCH* transfers effectively across different PTQ methods, with particularly consistent improvements under QuIP.

Table 8: Effect of the Stage 2 optimization budget under 2-bit GPTQ.

| Model | Stage 1 Only | Stage 2 Optimization Steps | | | | | |
|---|---|---|---|---|---|---|---|
| | | 10 | 50 | 60 (Ours) | 100 | 200 | 500 |
| **WikiText-2 PPL ↓** | | | | | | | |
| Llama-3.1-8B | 347.6 | 107.5 | 86.92 | 85.37 | 83.67 | 83.89 | **82.38** |
| Qwen3-8B | 87.64 | 34.21 | 28.61 | 28.37 | 27.90 | 27.68 | **27.49** |
| Gemma-2-9B | 34.64 | 23.45 | 22.52 | 22.51 | 22.44 | 22.39 | **22.36** |
| **Average Zero-Shot Accuracy (%) ↑** | | | | | | | |
| Llama-3.1-8B | 39.20 | 40.28 | 41.29 | 41.33 | 41.36 | 41.45 | **41.61** |
| Qwen3-8B | 47.16 | 49.20 | 49.91 | 49.97 | **50.10** | 49.91 | 49.87 |
| Gemma-2-9B | 52.36 | 54.71 | 55.01 | 54.99 | 54.93 | **55.02** | 54.83 |

Table 9: Effect of the Stage 2 optimization budget under 2-bit AWQ.

| Model | Stage 1 Only | Stage 2 Optimization Steps | | | | | |
|---|---|---|---|---|---|---|---|
| | | 10 | 50 | 60 (Ours) | 100 | 200 | 500 |
| **WikiText-2 PPL ↓** | | | | | | | |
| Llama-3.1-8B | 2,632.33 | 850.28 | 403.22 | 382.60 | 352.61 | 335.95 | **324.13** |
| Qwen3-8B | 245.83 | 50.24 | 41.53 | 41.24 | 40.98 | 40.93 | **40.84** |
| Gemma-2-9B | 98.02 | 39.47 | 33.97 | 33.78 | 33.50 | 33.32 | **33.20** |
| **Average Zero-Shot Accuracy (%) ↑** | | | | | | | |
| Llama-3.1-8B | 34.28 | 35.26 | 36.75 | **37.15** | 36.81 | 36.61 | 36.88 |
| Qwen3-8B | 41.32 | 46.70 | 48.02 | 48.23 | **48.46** | 48.32 | 48.13 |
| Gemma-2-9B | 43.03 | 49.69 | 51.13 | **51.28** | 51.22 | 51.19 | 51.03 |

Table 10: Effect of the Stage 2 optimization budget under 2-bit QuIP.

| Model | Stage 1 Only | Stage 2 Optimization Steps | | | | | |
|---|---|---|---|---|---|---|---|
| | | 10 | 50 | 60 (Ours) | 100 | 200 | 500 |
| **WikiText-2 PPL ↓** | | | | | | | |
| Llama-3.1-8B | 124.94 | 44.20 | 40.43 | 40.34 | 40.15 | 40.01 | **40.00** |
| Qwen3-8B | 49.43 | 31.18 | 28.77 | 28.59 | 28.37 | 28.30 | **28.24** |
| Gemma-2-9B | 21.08 | 17.65 | 17.33 | 17.31 | 17.29 | 17.29 | **17.28** |
| **Average Zero-Shot Accuracy (%) ↑** | | | | | | | |
| Llama-3.1-8B | 39.42 | 43.40 | **44.87** | 44.74 | 44.68 | 44.57 | 44.55 |
| Qwen3-8B | 45.46 | 49.17 | 50.13 | 50.21 | 50.04 | 50.30 | **50.33** |
| Gemma-2-9B | 58.51 | 59.49 | 59.87 | 59.81 | **59.91** | 59.83 | 59.89 |

# B Extended Ablation Study

## B.1 Effect of Stage 2 Optimization Budget

To assess the optimization budget required for Stage 2 compensation regulation, we compared Stage 1-only performance with Stage 2 optimization using 10–500 steps under GPTQ, AWQ, and QuIP. As shown in Tables 8–10, Stage 2 substantially improves both WikiText-2 PPL and average zero-shot accuracy across all three PTQ methods and models. Most of the gains occur within the first 60 steps: for example, under GPTQ, the Llama-3.1-8B PPL falls from 347.6 after Stage 1 alone to 85.37 at 60 steps, while its zero-shot accuracy rises from 39.20% to 41.33%. Most of the PPL improvement is achieved within 60 steps, after which further optimization yields diminishing returns. The remaining gap to 500 steps is small for most settings (*e.g.,* 0.88 and 0.15 for Qwen3-8B and Gemma-2-9B under GPTQ), although a larger gap of 58.47 remains for Llama-3.1-8B under AWQ. Zero-shot accuracy also largely stabilizes by 60 steps, with the best step varying across models

Table 11: Sensitivity to the number of C4 calibration sequences under 2-bit GPTQ.

| Samples | Llama-3.1-8B | | Qwen3-8B | | Gemma-2-9B | |
|---|---|---|---|---|---|---|
| | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ |
| 32 | 85.72 | 41.22 | 28.51 | **50.11** | 22.55 | **55.05** |
| 64 | 86.06 | **41.90** | **28.21** | 49.98 | 22.53 | 55.01 |
| **128 (Ours)** | 85.37 | 41.33 | 28.37 | 49.97 | 22.51 | 54.99 |
| 256 | **84.87** | 41.73 | 28.33 | 49.82 | **22.47** | 55.01 |

Table 12: Sensitivity to the number of C4 calibration sequences under 2-bit AWQ.

| Samples | Llama-3.1-8B | | Qwen3-8B | | Gemma-2-9B | |
|---|---|---|---|---|---|---|
| | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ |
| 32 | 409.14 | 37.29 | 42.70 | **48.35** | 34.32 | 50.98 |
| 64 | 401.74 | 36.80 | 41.59 | 48.21 | 34.06 | 51.10 |
| **128 (Ours)** | 382.60 | 37.15 | **41.24** | 48.23 | 33.78 | **51.28** |
| 256 | **361.45** | **37.30** | 41.40 | **48.35** | **33.61** | 51.23 |

Table 13: Sensitivity to the number of C4 calibration sequences under 2-bit QuIP.

| Samples | Llama-3.1-8B | | Qwen3-8B | | Gemma-2-9B | |
|---|---|---|---|---|---|---|
| | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ |
| 32 | 40.96 | 44.53 | 28.81 | **50.27** | 17.39 | 59.84 |
| 64 | 40.47 | 44.68 | 28.64 | 49.98 | 17.33 | 59.87 |
| **128 (Ours)** | 40.34 | 44.74 | 28.59 | 50.21 | 17.31 | 59.81 |
| 256 | **40.27** | **44.76** | **28.56** | 50.23 | **17.30** | **59.99** |

and PTQ methods. For instance, the 60-step setting gives the highest accuracy for two of the three models under AWQ, while the best results under GPTQ and QuIP are distributed across different step counts. These results indicate that additional steps can provide modest, method- and model-dependent gains, but do not yield a consistent improvement across all metrics. We therefore use 60 steps as the default, achieving substantial gains over Stage 1 alone while keeping the Stage 2 optimization budget limited.

### B.2 Impact of Calibration Dataset Size

To assess sensitivity to calibration-set size, we varied the number of C4 sequences from 32 to 64, 128, and 256, and repeated the two-stage optimization under GPTQ, AWQ, and QuIP. As shown in Tables 11–13, increasing the calibration set does not produce a consistent monotonic change in both WikiText-2 PPL and average zero-shot accuracy. Under GPTQ, for example, 128 samples give the lowest PPL for Llama-3.1-8B (85.37), while 256 samples give the lowest PPL for Qwen3-8B and Gemma-2-9B (28.33 and 22.47, respectively). The best zero-shot accuracy also varies across models and sample counts; the 128-sample setting is close to the best result for each model, with differences of at most 0.57 percentage points.

The same pattern holds under AWQ and QuIP. With AWQ, 256 samples yield the lowest PPL for all three models, but the PPL gains over 128 samples are modest for Qwen3-8B and Gemma-2-9B (0.16 and 0.17), and zero-shot accuracy changes by no more than 0.30 percentage points across sample counts for these models. Under QuIP, moving from 128 to 256 samples reduces PPL by 0.07, 0.03, and 0.01 for Llama-3.1-8B, Qwen3-8B, and Gemma-2-9B, respectively; accuracy changes by at most 0.18 percentage points. Thus, larger calibration sets can improve selected metrics, but the gains are small and do not consistently extend to both PPL and accuracy. We use 128 sequences as the default, which provides competitive performance across models and PTQ methods without incurring the additional calibration cost of larger sets.

Table 14: Ablation study of compensation structure and regulation granularity under 2-bit AWQ quantization.

| Compensation Structure | Regulation Granularity | Llama-3.1-8B PPL ↓ | Llama-3.1-8B Acc. ↑ | Qwen3-8B PPL ↓ | Qwen3-8B Acc. ↑ | Gemma-2-9B PPL ↓ | Gemma-2-9B Acc. ↑ |
|---|---|---|---|---|---|---|---|
| **Reference** | | | | | | | |
| None | None | 5.29e5 | 35.39 | 6.57e4 | 35.32 | 1.07e4 | 39.13 |
| **Compensation Structure** | | | | | | | |
| Low-rank ($r = 32$) | Layer | 1,139.9 | 35.62 | 186.8 | 38.20 | 55.06 | 46.03 |
| Channel | Layer | 451.4 | 36.77 | 48.14 | 47.57 | 37.17 | 50.65 |
| **Group** | Layer | **382.6** | **37.15** | **41.24** | **48.23** | **33.78** | **51.28** |
| **Regulation Granularity** | | | | | | | |
| Group | None | 2.63e3 | 34.28 | 245.8 | 41.32 | 98.02 | 43.03 |
| Group | Global | 1.46e3 | 34.49 | 71.90 | 45.68 | 49.13 | 49.68 |
| Group | Block | 811.1 | 34.81 | 50.40 | 47.00 | 44.59 | 49.86 |
| Group | **Layer** | **382.6** | **37.15** | **41.24** | **48.23** | **33.78** | **51.28** |

Table 15: Ablation study of compensation structure and regulation granularity under 2-bit GPTQ quantization.

| Compensation Structure | Regulation Granularity | Llama-3.1-8B PPL ↓ | Llama-3.1-8B Acc. ↑ | Qwen3-8B PPL ↓ | Qwen3-8B Acc. ↑ | Gemma-2-9B PPL ↓ | Gemma-2-9B Acc. ↑ |
|---|---|---|---|---|---|---|---|
| **Reference** | | | | | | | |
| None | None | 292.1 | 36.81 | 59.34 | 39.46 | 32.77 | 49.35 |
| **Compensation Structure** | | | | | | | |
| Low-rank ($r = 32$) | Layer | 173.7 | 37.24 | 48.27 | 41.46 | 29.12 | 51.02 |
| Channel | Layer | 100.0 | 41.01 | 28.96 | 49.68 | 22.97 | 54.75 |
| **Group** | Layer | **85.37** | **41.33** | **28.37** | **49.97** | **22.51** | **54.99** |
| **Regulation Granularity** | | | | | | | |
| Group | None | 347.6 | 39.20 | 87.64 | 47.16 | 34.64 | 52.36 |
| Group | Global | 124.3 | 39.32 | 40.88 | 48.31 | 24.57 | 54.27 |
| Group | Block | 131.0 | 40.24 | 30.44 | 49.32 | 23.85 | **55.07** |
| Group | **Layer** | **85.37** | **41.33** | **28.37** | **49.97** | **22.51** | 54.99 |

### B.3 Extended Ablations Across PTQ Methods

To assess whether our ablation findings generalize across PTQ methods, we repeated the study using 2-bit AWQ and GPTQ. As shown in Tables 14 and 15, group-wise compensation achieves the lowest PPL and highest accuracy among the tested compensation structures for all three models under both methods. Compared with channel-wise compensation, it reduces PPL by 2.0%–15.2% and improves accuracy by 0.24–0.66 percentage points. Layer-wise regulation also generally outperforms coarser regulation. It achieves the lowest PPL for all models under both PTQ methods and the highest accuracy in five of the six model-method pairs. The only exception is Gemma-2-9B under GPTQ, where block-wise regulation gives slightly higher accuracy (55.07% vs. 54.99%). These results demonstrate the effectiveness of group-wise compensation and layer-wise regulation across different PTQ methods.

## C Correction Capacity and Structural Efficiency

### C.1 Rank Properties of Vector Compensation

We analyze how vector compensation can produce a high-rank correction with relatively few parameters. Given a quantized weight matrix $\mathbf{W}_q \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, LoRA-based methods correct the weight

Table 16: Effect of compensation rank for representative low-rank methods under 2-bit GPTQ.

| Method | Rank | Llama-3.1-8B | | Qwen3-8B | | Gemma-2-9B | |
|---|---|---|---|---|---|---|---|
| | | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ |
| $L^2$QER | 32 | 187.6 | 37.87 | 47.55 | 41.13 | 30.20 | 50.82 |
| | 64 | 184.2 | 37.90 | 42.14 | 42.32 | 26.75 | 51.55 |
| | 128 | 138.7 | 39.44 | 35.86 | 43.29 | 23.43 | 53.15 |
| QERA-A | 32 | 175.7 | 38.36 | 45.28 | 41.93 | 28.96 | 50.64 |
| | 64 | 168.5 | 37.95 | 39.74 | 42.24 | 25.97 | 52.51 |
| | 128 | 125.0 | 38.98 | 34.02 | 44.23 | 23.00 | 53.84 |
| QERA-E | 32 | 172.2 | 37.81 | 48.20 | 40.69 | 30.14 | 50.91 |
| | 64 | 160.7 | 38.14 | 42.40 | 42.00 | 28.13 | 51.70 |
| | 128 | 137.9 | 38.10 | 35.68 | 43.04 | 24.79 | 53.33 |
| **PATCH** | – | **85.37** | **41.33** | **28.37** | **49.97** | **22.51** | **54.99** |

as $\widehat{\mathbf{W}}_{\mathrm{LoRA}} = \mathbf{W}_q + \mathbf{AB}$, where $\mathbf{A} \in \mathbb{R}^{d_{\mathrm{out}} \times r}$ and $\mathbf{B} \in \mathbb{R}^{r \times d_{\mathrm{in}}}$. The correction $\Delta\mathbf{W}_{\mathrm{LoRA}} = \mathbf{AB}$ therefore satisfies $\mathrm{rank}(\Delta\mathbf{W}_{\mathrm{LoRA}}) \leq r$ and is explicitly constrained by the adapter rank specified in advance.

By contrast, channel-wise vector compensation applies a diagonal matrix $\mathbf{D} = \mathrm{Diag}(\boldsymbol{\alpha})$ to the quantized weight, giving $\widehat{\mathbf{W}}_{\mathrm{vec}} = \mathbf{DW}_q$. The resulting correction is $\Delta\mathbf{W} = (\mathbf{D} - \mathbf{I})\mathbf{W}_q$.

**Proposition 1.** If $\mathbf{W}_q$ has maximum rank and $\alpha_i \neq 1$ for every output channel $i$, then the correction produced by channel-wise vector compensation also has maximum rank.

*Proof.* Because $\alpha_i \neq 1$ for every $i$, the diagonal matrix $\mathbf{D} - \mathbf{I}$ is nonsingular. Left multiplication by a nonsingular matrix preserves rank; thus,

$$\mathrm{rank}(\Delta\mathbf{W}) = \mathrm{rank}((\mathbf{D} - \mathbf{I})\mathbf{W}_q) = \mathrm{rank}(\mathbf{W}_q) = \min(d_{\mathrm{out}}, d_{\mathrm{in}}).$$

Therefore, channel-wise vector compensation can produce a maximum-rank correction using only $d_{\mathrm{out}}$ scaling parameters. *PATCH* extends channel-wise vector compensation to fine-grained quantization groups. Partition $\mathbf{W}_q$ along its input dimension into $G$ groups: $\mathbf{W}_q = [\mathbf{W}_q^{(1)}, \ldots, \mathbf{W}_q^{(G)}]$. Let $\mathbf{M}$ denote the compensation matrix defined in Section 4.1, with entries $M_{c,i} = m_{c,g(i)}$. The *PATCH* correction can then be written group-wise as

$$\Delta\mathbf{W}_{\mathrm{PATCH}} = [(\mathbf{M}_1 - \mathbf{I})\mathbf{W}_q^{(1)}, \ldots, (\mathbf{M}_G - \mathbf{I})\mathbf{W}_q^{(G)}],$$

where $\mathbf{M}_g = \mathrm{Diag}(m_{1,g}, \ldots, m_{d_{\mathrm{out}},g})$. Equivalently, using the notation of Section 4.1,

$$\Delta\mathbf{W}_{\mathrm{PATCH}} = (\mathbf{M} - \mathbf{1}) \odot \mathbf{W}_q.$$

*PATCH* extends channel-wise vector compensation by assigning an independent scaling coefficient to each output-channel–group pair. Setting $m_{c,g} = \alpha_c$ for every group $g$ recovers the channel-wise correction $(\mathbf{D} - \mathbf{I})\mathbf{W}_q$. Thus, the PATCH parameterization retains channel-wise vector compensation as a special case while allowing group-specific corrections within each output channel. This additional granularity provides greater flexibility to model heterogeneous quantization errors across input groups, without imposing an explicit rank constraint on the correction.

### C.2 Effect of Low-Rank Compensation Rank

To examine whether increasing the rank closes the performance gap between low-rank compensation and *PATCH*, we evaluated $L^2$QER, QERA-A, and QERA-E at ranks 32, 64, and 128 under 2-bit GPTQ. As shown in Table 16, increasing the rank generally improves PPL, although zero-shot accuracy does not increase consistently. Even at rank 128, the low-rank methods remain behind *PATCH* on all three models. At rank 128, the best low-rank baselines achieve PPLs of 125.0, 34.02, and 23.00, compared with 85.37, 28.37, and 22.51 for *PATCH*. Their best zero-shot accuracies are also lower than those of *PATCH* by 1.89, 5.74, and 1.15 percentage points, respectively. These results indicate that simply increasing the rank of the evaluated low-rank corrections does not match *PATCH*'s performance, which uses a fine-grained compensation parameterization without an explicit rank constraint.

Table 17: Parameter–performance trade-off between QERA-E and PATCH across 2-bit PTQ methods.

| Method | Rank | Params. (M)† | Ratio‡ | GPTQ | | AWQ | | QuIP | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ |
| QERA-E | 32 | 83.9–108.0 | 1.5–1.6× | 83.51 | 43.14 | 1,033.54 | 40.27 | 100.01 | 44.73 |
| | 64 | 167.8–216.1 | 3.0–3.2× | 77.08 | 43.95 | 486.18 | 41.27 | 91.63 | 45.11 |
| | 128 | 335.5–432.1 | 6.0–6.5× | 66.12 | 44.82 | 236.08 | 42.79 | 70.86 | 46.39 |
| | 256 | 671.1–864.3 | 12.0–12.9× | 47.16 | 46.98 | **100.08** | **45.86** | 38.84 | 48.68 |
| **PATCH** | – | **55.7–66.9** | **1.0×** | **45.42** | **48.76** | 152.54 | 45.55 | **28.75** | **51.59** |

† Additional parameters (M), shown as the range across the three models. ‡ Parameter count relative to *PATCH*.

Table 18: Sensitivity to compensation group size under 2-bit GPTQ.

| Compensation Group Size | Llama-3.1-8B | | Qwen3-8B | | Gemma-2-9B | |
|---|---|---|---|---|---|---|
| | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ |
| 512 | 94.02 | **41.35** | **28.28** | 50.05 | 22.66 | 55.02 |
| 256 | 95.62 | 41.30 | 28.36 | **50.54** | 22.58 | **55.18** |
| **128 (Ours)** | **85.37** | 41.33 | 28.37 | 49.97 | **22.51** | 54.99 |

Table 19: Effect of compensation bias under 2-bit GPTQ.

| Variant | Llama-3.1-8B | | Qwen3-8B | | Gemma-2-9B | |
|---|---|---|---|---|---|---|
| | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ |
| PATCH w/o $\beta$ | 99.38 | **41.52** | 28.62 | 49.20 | 23.26 | 54.49 |
| **PATCH** | **85.37** | 41.33 | **28.37** | **49.97** | **22.51** | **54.99** |

### C.3 Parameter Budget and Performance Trade-offs

Table 17 compares QERA-E and *PATCH* in terms of compensation parameters and performance across three 2-bit PTQ methods. *PATCH* uses only 55.7–66.9M parameters, whereas QERA-E requires 83.9–108.0M even at rank 32 and 671.1–864.3M at rank 256. Despite this smaller parameter budget, *PATCH* achieves the best GPTQ and QuIP results, improving average accuracy over QERA-E at rank 256 by 1.78 and 2.91 percentage points, respectively, while also lowering PPL. Under AWQ, rank-256 QERA-E achieves the best overall performance, but requires 12.0–12.9× the parameter budget of *PATCH*; *PATCH* remains competitive, with only a 0.31-point accuracy gap. Overall, *PATCH* offers a favorable parameter–performance trade-off, particularly under GPTQ and QuIP, without requiring a rank-dependent increase in compensation parameters.

## D Stage 1 Fine-Grained Compensation Analysis

### D.1 Effect of Compensation Group Size

Table 18 evaluates the trade-off between compensation granularity and the number of compensation coefficients. A group size of 128 assigns an independent compensation factor to each quantization group, whereas sizes of 256 and 512 share each factor across two and four quantization groups, respectively. Although group size 128 does not yield the highest accuracy for every model, it achieves the lowest PPL on Llama-3.1-8B and Gemma-2-9B, with competitive accuracy across all three models. Larger groups reduce the number of compensation coefficients by sharing them across multiple quantization groups, but provide coarser compensation. We therefore use a group size of 128 to retain quantization-group-level compensation granularity.

Table 20: Effect of Stage 2 regulation across Stage 1 compensation structures and 2-bit PTQ methods.

| PTQ | Compensation Structure | WikiText-2 PPL | | | Average Zero-Shot Accuracy (%) | | |
|---|---|---|---|---|---|---|---|
| | | Before | After | Reduction (%) ↑ | Before | After | Gain (pp) ↑ |
| GPTQ | Low-rank ($r = 32$) | 83.51 | 83.70 | −0.22 | 43.14 | 43.24 | +0.10 |
| | Channel | 154.05 | 50.64 | +67.13 | 45.61 | 48.48 | +2.87 |
| | Group | 156.63 | **45.42** | +71.00 | 46.24 | **48.76** | +2.52 |
| AWQ | Low-rank ($r = 32$) | 1,033.54 | 460.59 | +55.44 | 40.27 | 39.95 | −0.32 |
| | Channel | 2,046.00 | 178.90 | +91.26 | 38.88 | 45.00 | +6.12 |
| | Group | 992.04 | **152.54** | +84.62 | 39.54 | **45.55** | +6.01 |
| QuIP | Low-rank ($r = 32$) | 100.01 | 72.18 | +27.83 | 44.73 | 45.10 | +0.37 |
| | Channel | 83.35 | 31.17 | +62.61 | 47.46 | 50.72 | +3.26 |
| | Group | 65.14 | **28.75** | +55.87 | 47.80 | **51.59** | +3.79 |

Table 21: Stage 2 regulation by linear layer or within-layer weight group under 2-bit PTQ.

| PTQ | Regulation Granularity | Llama-3.1-8B | | Qwen3-8B | | Gemma-2-9B | |
|---|---|---|---|---|---|---|---|
| | | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ |
| GPTQ | Layer | **85.37** | **41.33** | **28.37** | **49.97** | **22.51** | **54.99** |
| | Layer× Group | 96.53 | 39.59 | 31.07 | 46.42 | 24.00 | 53.68 |
| AWQ | Layer | **382.6** | **37.15** | **41.24** | **48.23** | **33.78** | **51.28** |
| | Layer× Group | 1,220.0 | 35.79 | 65.99 | 42.63 | 41.89 | 49.21 |
| QuIP | Layer | **40.34** | **44.74** | **28.59** | **50.21** | **17.31** | **59.81** |
| | Layer× Group | 46.77 | 42.84 | 38.04 | 46.87 | 17.89 | 59.06 |

### D.2 Contribution of the Compensation Bias

Table 19 evaluates the contribution of the compensation bias $\beta$, with the group-wise scale structure and layer-wise Stage 2 gating held fixed. Removing the bias increases PPL on all three models: from 85.37 to 99.38 on Llama-3.1-8B, from 28.37 to 28.62 on Qwen3-8B, and from 22.51 to 23.26 on Gemma-2-9B. The full *PATCH* configuration also improves zero-shot accuracy on Qwen3-8B and Gemma-2-9B by 0.77 and 0.50 percentage points, respectively. On Llama-3.1-8B, removing the bias gives a marginally higher accuracy (41.52% vs. 41.33%), despite its higher PPL. Overall, the bias consistently reduces perplexity and improves accuracy on two of the three models, with a small accuracy trade-off on Llama-3.1-8B.

## E Stage 2 Compensation Regulation Analysis

### E.1 Effect of Stage 2 Regulation Across Compensation Structures

Table 20 reports the effect of applying the same layer-wise Stage 2 regulation to low-rank, channel-wise, and group-wise compensation under GPTQ, AWQ, and QuIP. Results are averaged across the three models. Regulation substantially reduces PPL and improves zero-shot accuracy for channel-wise and group-wise compensation across all three PTQ methods. For group-wise compensation, PPL decreases by 71.00%, 84.62%, and 55.87% under GPTQ, AWQ, and QuIP, respectively, while accuracy increases by 2.52, 6.01, and 3.79 percentage points. The low-rank variant benefits less consistently: under GPTQ, its PPL slightly increases after regulation, and under AWQ its accuracy decreases slightly.

Among the regulated variants, group-wise compensation achieves the lowest PPL and highest accuracy for each PTQ method. Compared with regulated channel-wise compensation, it reduces PPL by 5.22, 26.36, and 2.42 under GPTQ, AWQ, and QuIP, respectively, and improves accuracy by 0.28, 0.55, and 0.87 percentage points. These results indicate that Stage 2 regulation provides consistent

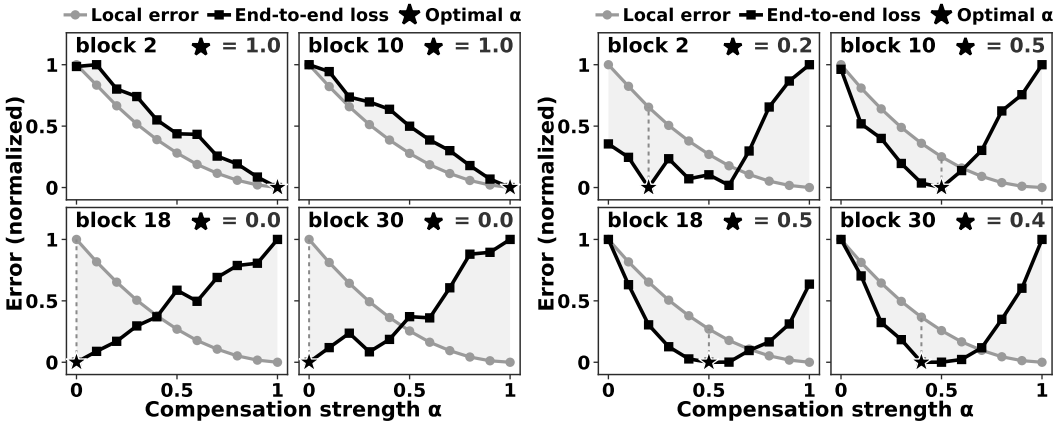
gains when paired with group-wise compensation, and that the combination performs best across the evaluated PTQ backbones.

### E.2 Effect of Stage 2 Compensation Regulation Granularity

Table 21 compares layer-wise Stage 2 regulation with Layer-wise $\times$ Group regulation, while keeping the Stage 1 group-wise compensation structure fixed. Layer-wise regulation applies one coefficient to each linear layer, whereas Layer-wise $\times$ Group regulation uses separate coefficients for the weight groups within each layer. Layer-wise regulation achieves lower PPL and higher zero-shot accuracy across all three models and PTQ methods. The largest difference occurs for Qwen3-8B under AWQ: PPL decreases from 65.99 to 41.24, and accuracy increases by 5.60 percentage points. These results show that applying separate regulation coefficients to groups within a layer is less effective than using a single coefficient for the entire layer in the evaluated settings.

## F Additional Analysis of Motivation

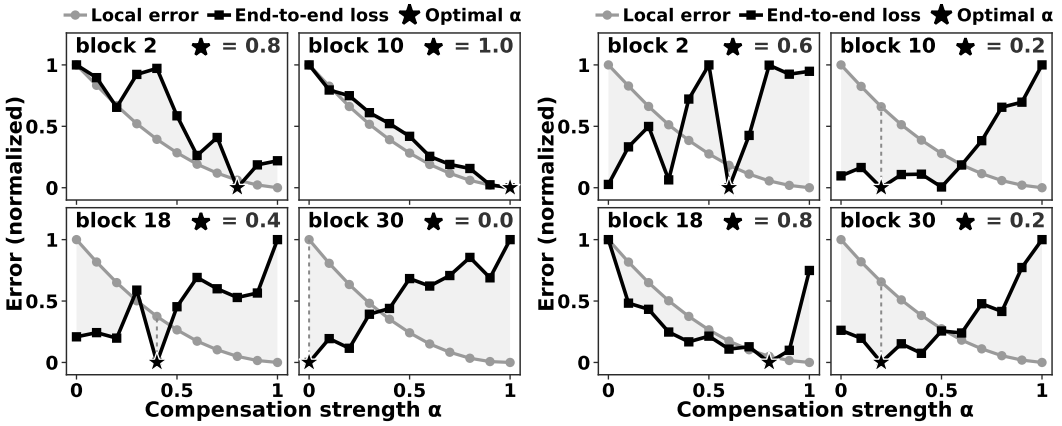
This section provides additional analyses of our motivation (Section 3.2). Across Gemma-2-9B, Llama-3.1-8B, and Qwen3-8B, we observe a mismatch between the compensation strengths preferred by the local and end-to-end objectives. In the fully quantized setting, where quantization errors propagate through the network, the end-to-end optimal $\alpha$ varies across transformer blocks. In the isolated setting, where such propagation is absent, the local and end-to-end objectives can still prefer different optimal $\alpha$ values. These results show that the mismatch identified in motivation (Section 3.2) is consistently observed across models and transformer blocks.




(a) Fully quantized setting



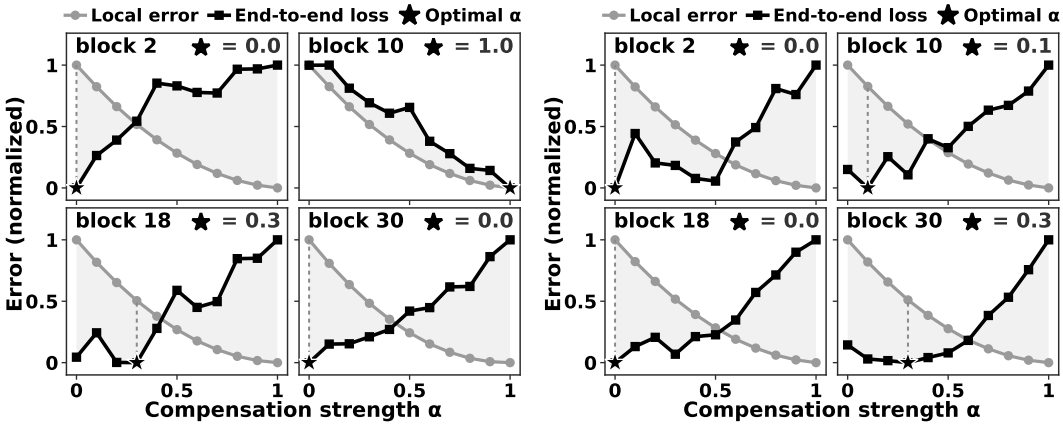

(b) Isolated setting




(a) Fully quantized setting




(b) Isolated setting

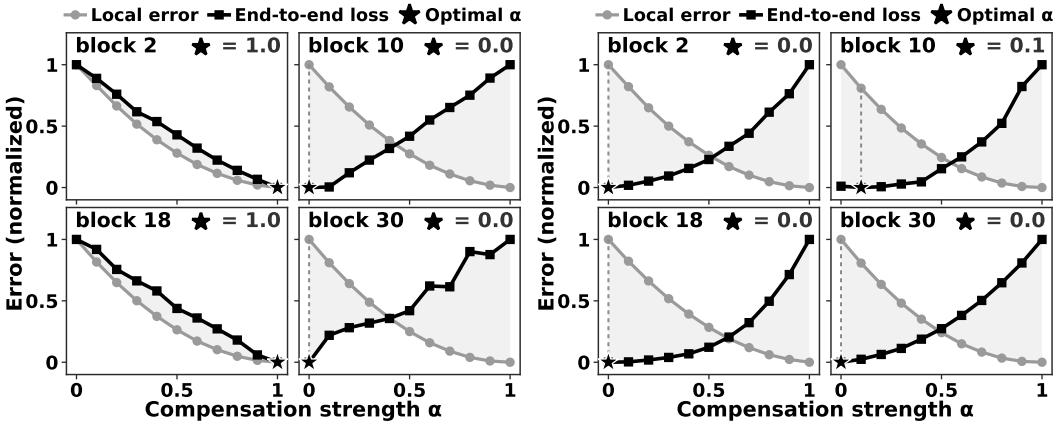



(a) Fully quantized setting




(b) Isolated setting

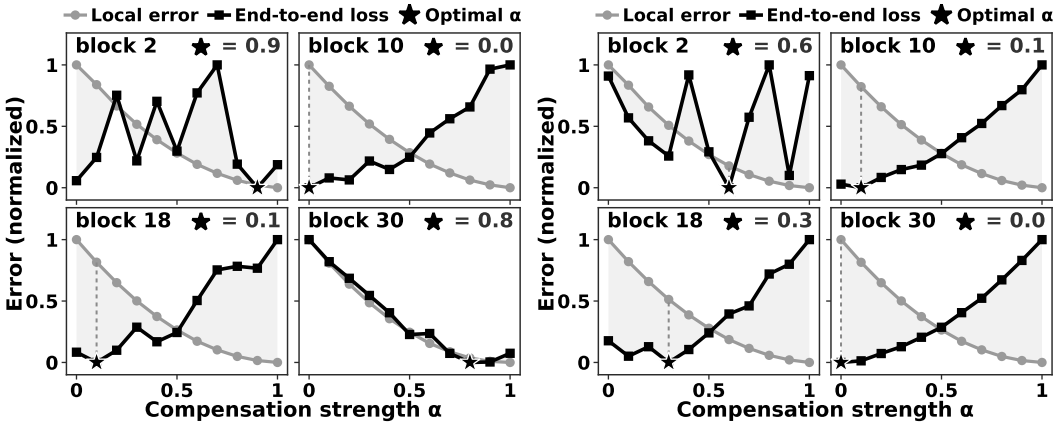
Figure 6: Propagation-aware compensation analysis for `q_proj` across transformer blocks under 2-bit GPTQ. Rows from top to bottom: Llama-3.1-8B, Qwen3-8B, and Gemma-2-9B.




(a) Fully quantized setting



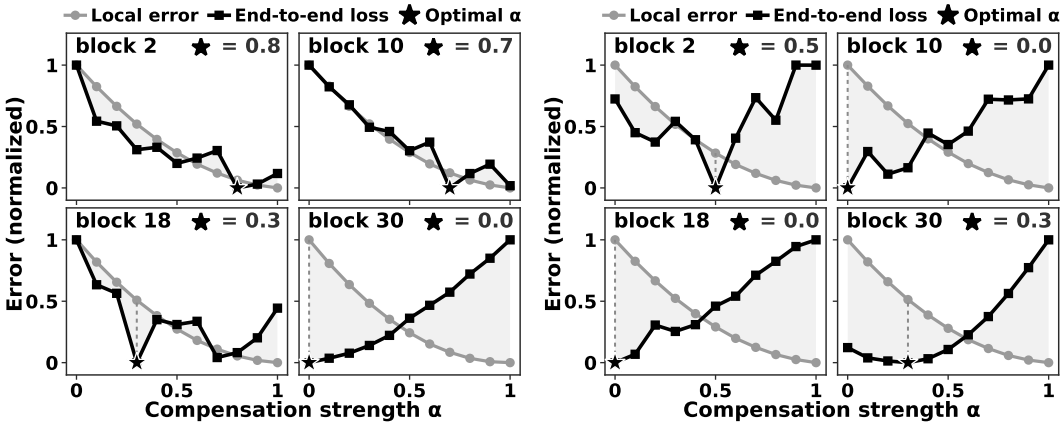

(b) Isolated setting




(a) Fully quantized setting




(b) Isolated setting

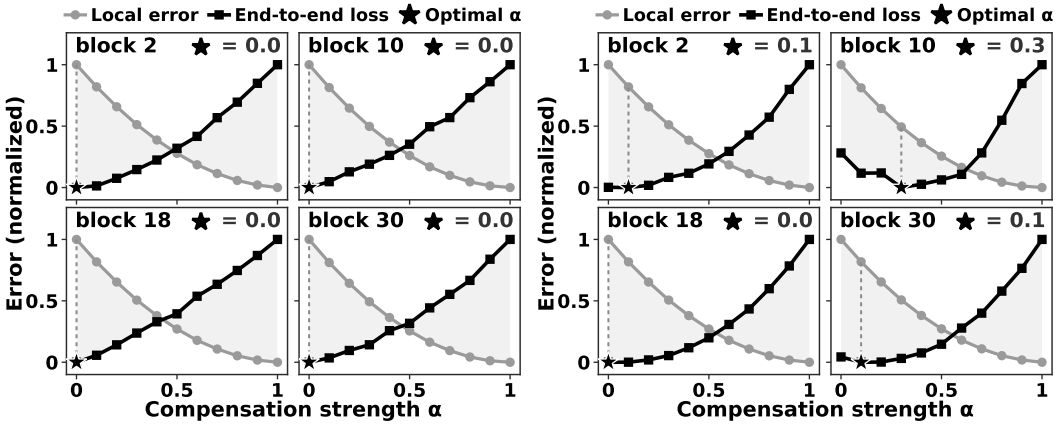



(a) Fully quantized setting




(b) Isolated setting

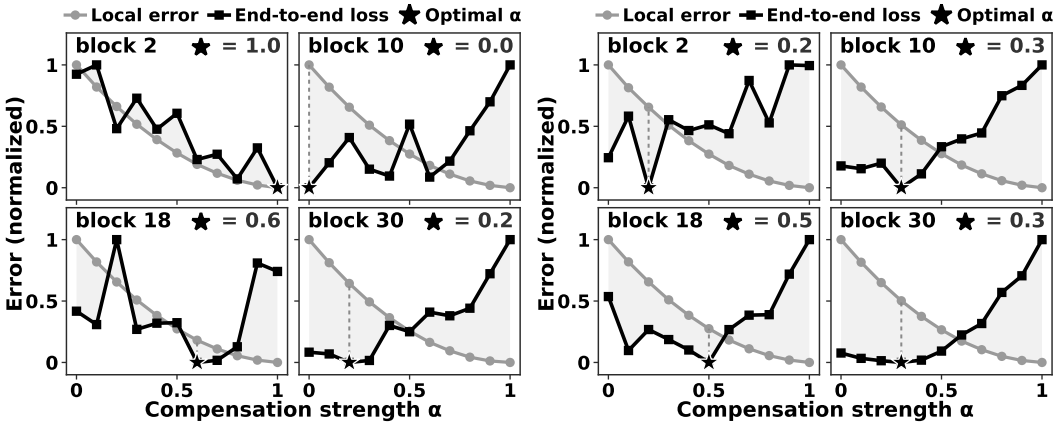
Figure 7: Propagation-aware compensation analysis for `k_proj` across transformer blocks under 2-bit GPTQ. Rows from top to bottom: Llama-3.1-8B, Qwen3-8B, and Gemma-2-9B.




(a) Fully quantized setting



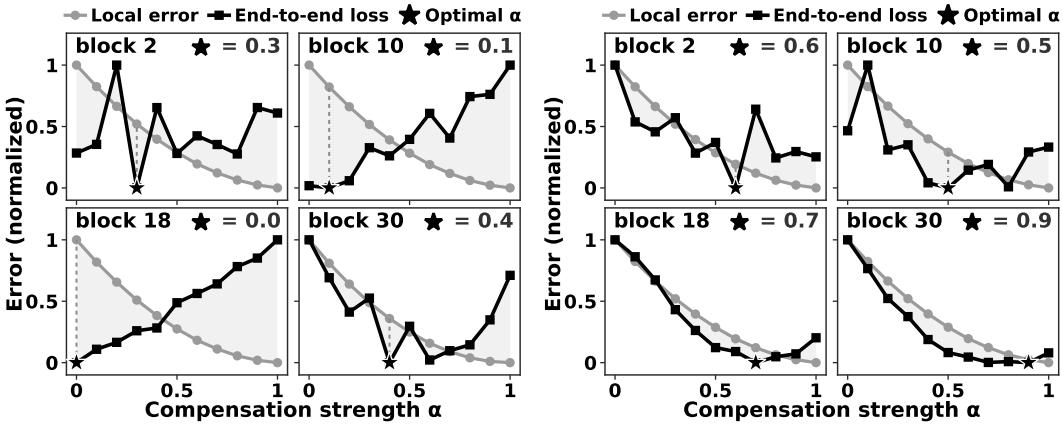

(b) Isolated setting




(a) Fully quantized setting




(b) Isolated setting




(a) Fully quantized setting




(b) Isolated setting

Figure 8: Propagation-aware compensation analysis for `v_proj` across transformer blocks under 2-bit GPTQ. Rows from top to bottom: Llama-3.1-8B, Qwen3-8B, and Gemma-2-9B.