SIGNROUNDV2: TOWARD CLOSING THE PERFORMANCE GAP IN EXTREMELY LOW-BIT POST-TRAINING QUANTIZATION FOR LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Extremely low-bit quantization is critical for efficiently deploying Large Language Models (LLMs), yet it often leads to severe performance degradation at 2 bits and even at 4 bits (e.g., MXFP4). We present SignRoundV2, a post-training quantization framework designed to maintain high performance even under aggressive compression. SignRoundV2 introduces (1) a simple yet efficient adaptive mixed-precision strategy that leverages gradient information and quantization-induced reconstruction errors to guide layer-wise bit allocation, and (2) a set of lightweight techniques, including loss filtering and a pre-tuning scale search, to improve tuning effectiveness in extremely low-bit regimes. Our approach takes a significant step toward closing the performance gap between quantized and full-precision models. Experimental results across diverse LLMs demonstrate that SignRoundV2 achieves near-lossless performance in mixed MXFP settings, narrowing the gap to $\sim$1% at an average of 4.5 bits, while substantially improving accuracy in challenging 2-bit weight-only quantization.

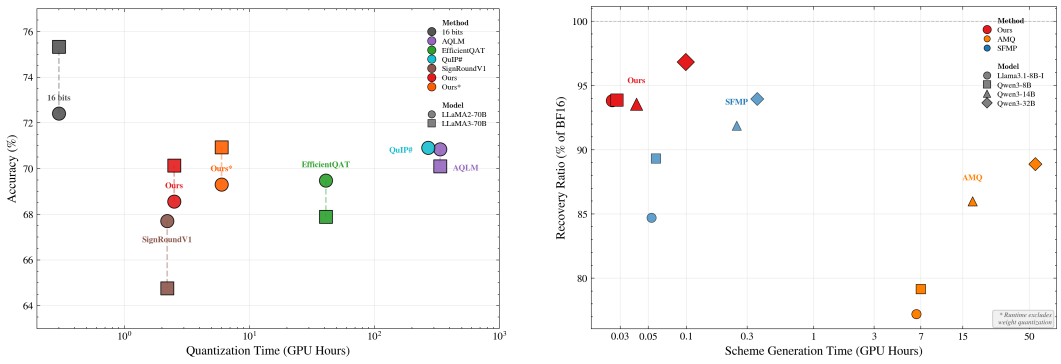
1 INTRODUCTION

The advent of Large Language Models (LLMs) such as GPT (Brown et al., 2020), LLaMA (Touvron et al., 2023a; Grattafiori et al., 2024), Qwen (Bai et al., 2023; Yang et al., 2025) and DeepSeek (Liu et al., 2024a) has marked a major shift in modern artificial intelligence. As parameter counts scale from billions to hundreds of billions, these models have demonstrated unprecedented capabilities in reasoning, coding, multimodal understanding, and autonomous agent behaviors. However, such scaling comes at the cost of dramatically increased memory consumption, bandwidth pressure, and inference latency, posing significant barriers for real-world deployment, especially in resource-constrained environments such as consumer GPUs, CPUs, and edge devices.

To alleviate these limitations, post-training quantization (PTQ) (Xiao et al., 2023; Frantar et al., 2022; Lin et al., 2024b; Cheng et al., 2024) has emerged as one of the most practical approaches because it avoids costly re-training and can be applied to a wide range of pretrained LLMs. By compressing weights and activations to low-bit representations, PTQ yields substantial reductions in memory footprint and hardware cost. In particular, the push toward extremely low-bit quantization, for example, sub-4-bit weight-only quantization and 4-bit activation quantization, has become a crucial enabler for democratizing LLM deployment.

Recent research has moved beyond conventional W4A16 and W8A8 schemes (where W and A denote weight and activation bit-widths) toward more expressive low-bit data types. For example, MXFP4 (Alliance, 2023) and NVFP4 (NVIDIA, 2024) are floating-point variants provided by modern accelerators, demonstrating that well-designed data types can maintain reasonable accuracy even under aggressive activation quantization. Despite these architectural advances, achieving high-accuracy recovery remains non-trivial, primarily due to the prevalent outliers in large-scale model. In parallel, weight-only quantization continues to push the limits of bit efficiency. In particular, BitNet (Wang et al., 2025; Wu et al., 2025) explores quantization down to approximately 1.58 bits, showing that extremely low-bit representations can still be learned with specialized training or distillation. However, these methods require training from scratch or costly post-training distillation, making them less suitable for scenarios where fast, scalable, and hardware-efficient PTQ is desired.

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.



(a) W2A16 results on Llama 2/3 70B

(b) Recovery ratio at 2.5 bits

Figure 1: Left: results on pure 2 bits (W2A16) for Llama 2/3 70B. Right: recovery ratio for adaptive bit assignments at 2.5 bits across four models.

To minimize the accuracy loss caused by quantization, adaptive mixed-precision quantization has emerged as a natural solution to the limitations of uniform quantization. By strategically allocating higher precision to sensitive layers while aggressively compressing more robust ones, it effectively balances model performance with computational efficiency (Zhou et al., 2018; Dong et al., 2019; Wang et al., 2019; You et al., 2024; Malinovskii et al., 2024). Therefore, the central challenge is the design of an accurate and efficient layer sensitivity metric to guide bit allocation. Early methods estimate sensitivity using Hessian-based metrics (Dong et al., 2019; You et al., 2024; Nie et al., 2026), or rely on layer-wise or model-wise output distortion measured on a small calibration dataset (Hubara et al., 2020). However, these heuristics often exhibit imperfect correlation when applied layer-wise, and model-wise approaches typically require slow calibration, especially for large language models (LLMs). In contrast, learning-based or reinforcement-learning-based approaches automate precision assignment (Lou et al., 2019; Wang et al., 2019; Lee et al., 2025), but at the cost of substantial compute overhead.

To address these challenges, we introduce SignRoundV2, an enhanced post-training quantization framework building upon SignRound (Cheng et al., 2024). Our method first utilizes a simple yet efficient adaptive mixed-precision strategy that integrates gradient information with quantization-induced deviations to capture both local parameter distortion and global task impact. By leveraging these reliable sensitivity signals, SignRoundV2 employs dynamic programming (Bellman, 1966) to efficiently determine the optimal layer-wise bit configuration for a given budget, substantially narrowing the performance gap between quantized and full-precision models even under aggressive compression. Second, motivated by the importance of quantization parameter initialization (Lin et al., 2024b; Ma et al., 2024; Bhalgat et al., 2020), we introduce a lightweight pre-tuning search to further improve performance, inspired by the importance matrix in llama.cpp (Gerganov, 2023). To better balance easy and hard samples, we further introduce a loss filtering mechanism that excludes high-loss outliers during tuning, preventing gradients from hard samples from degrading the learned quantization parameters.

Figure 1 compares our method to state-of-the-art baselines, showing that it achieves similar performance at significantly lower cost and avoids the issues associated with Quantization-Aware Training (QAT), which are discussed in Section 2. In summary, our contributions are three-fold.

- We introduce an efficient adaptive mixed-precision strategy that leverages gradient information and quantization-induced deviations to guide layer-wise bit allocation via dynamic programming, maintaining high performance even under aggressive compression.
- We employ simple yet effective techniques, such as loss filtering and a lightweight pre-tuning scale search, to improve accuracy in extremely low-bit regimes.
- We demonstrate that SignRoundV2 achieves competitive accuracy compared with recent high-cost QAT and vector-quantization methods in weight-only quantization, while requiring substantially lower quantization cost and maintaining strong performance in weight-activation scenarios.

2 Related Work

Quantization-Aware Training (QAT). QAT integrates learnable quantizers into the training loop and optimizes the task loss, often fine-tuning parameters beyond those introduced by low-precision operations, as in EfficientQAT (Chen et al., 2025), LLM-QAT (Liu et al., 2023b), DL-QAT (Ke et al., 2024), and BitDistiller (Du et al., 2024). Recent works (Chee et al., 2023; Tseng et al., 2024; Egiazarian et al., 2024) revisit vector quantization for extremely low-bit settings, but still rely on QAT to achieve high accuracy. Despite their effectiveness, QAT methods have several practical limitations. First, optimizing the task loss and updating non-quantizer parameters can make them sensitive to the fine-tuning data distribution, potentially causing forgetting or domain-specific overfitting (Luo et al., 2025; Kalajdzievski, 2024). Second, QAT often requires careful hyperparameter tuning and incurs substantially higher quantization costs than PTQ. Third, QAT typically requires more training data than PTQ for stable generalization, further increasing resource requirements.

Post-Training Quantization (PTQ). Post-training quantization (PTQ) avoids task-loss optimization and additional fine-tuning of model parameters and therefore offers a simple and resource-efficient pipeline for compressing LLMs (Nagel et al., 2019; Liu et al., 2021; Frantar & Alistarh, 2022; Yao et al., 2021; Xiao et al., 2023; Wei et al., 2023; Dettmers et al., 2022). Because LLM decoding is typically memory-bound, weight-only PTQ has become the dominant approach (Frantar et al., 2022; Lin et al., 2024b; Cheng et al., 2023; Yao et al., 2024; Badri & Shaji, 2023; Kim et al., 2023; Cheng et al., 2024) in LLM inference. However, existing weight-only methods still suffer from a significant accuracy drop at extremely low bit-widths (e.g., 2-bit). Meanwhile, joint weight–activation quantization has emerged as a complementary direction, with techniques such as rotation-based transformations (Chee et al., 2023; Ashkboos et al., 2024; Liu et al., 2024b; Lin et al., 2024a; Sun et al., 2024) proposed to address activation outliers, albeit at the cost of additional inference overhead.

Adaptive Mixed-Precision Quantization. Adaptive mixed-precision quantization assigns different bit-widths across layers or channels according to their sensitivity to quantization error. Early methods typically relied on second-order information, such as Hessian-based methods (e.g., HAWQ (Dong et al., 2019) and BAQ (Zhang et al., 2025)), or reinforcement learning, such as HAQ (Wang et al., 2019). However, they are costly for multi-billion-parameter LLMs due to expensive second-order computations or extensive policy evaluations. MixLLM (Zheng et al., 2024) combines first- and second-order metrics for channel-level precision, but its naive thresholding and neglect of activation distortion in weight-activation joint quantization can lead to suboptimal performance. Recent methods therefore focus on reducing the search space or avoiding search altogether. (Malinovskii et al., 2024) establish a linear relationship between layer-wise weight MSE and the resulting perplexity increase, and leverage it for data-free and non-uniform LLM quantization. AMQ (Lee et al., 2025) formulates mixed-precision assignment as an AutoML problem and uses search-space pruning and quantization proxies to accelerate the process, but remains relatively slow for large models. SFMP (Nie et al., 2026) avoids costly search with a fine-grained, hardware-friendly assignment strategy. MicroMix (Liu et al., 2025) similarly applies FP4/FP6/FP8 at the channel level using MXFP microscaling kernels. However, these methods often depend on specific hardware or specialized kernels. In practice, frameworks such as llama.cpp (Gerganov, 2023) use lightweight heuristics to assign mixed precision across different architectures. With the growing use of Mixture-of-Experts (MoE) models, recent methods such as MxMoE (Duanmu et al., 2025), MoQAE (Tao et al., 2025), and MoPEQ (Chitty-Venkata et al., 2025) further extend adaptive mixed-precision quantization to MoE models.

Rounding and Quantizer Optimization. Rounding critically affects the quality of quantized weights. AdaRound (Nagel et al., 2020) formulates weight rounding as a per-weight binary optimization problem, where a second-order Taylor expansion of the task loss approximates quantization-induced perturbations, and a layer-wise local reconstruction loss is minimized via continuous relaxation. This formulation has inspired subsequent works such as BRECQ (Li et al., 2021) and SignRound (Cheng et al., 2024). FlexRound (Lee et al., 2023) increases rounding flexibility through element-wise scaling, albeit at the cost of introducing numerous hyperparameters. Oscillation-Free training (Liu et al., 2023a) highlights instability arising from learnable rounding parameters. Activation-dependent rounding in AQuant (Li et al., 2022) reduces activation error but is not applicable to weight-only inference. SignRound jointly optimizes rounding and weight clipping using sign-based gradient descent, achieving substantial improvements.

3 METHODOLOGY

This section presents our methodology. We begin in Section 3.1 by introducing the sensitivity metric, *DeltaLoss*, which guides mixed-bit assignment. Section 3.2 then leverages DeltaLoss with dynamic programming to determine adaptive bit-widths for each layer. Section 3.3 describes the tuning strategy for quantization parameters, while Section 3.4 introduces our loss exclusion strategy. Finally, Section 3.5 summarizes the main hyperparameters and presents a practical trick for loss computation.

Before introducing our method, we briefly review the standard quantize-dequantize (QDQ) formulation and SignRoundV1 (Cheng et al., 2024). The QDQ operator for weights $\mathbf{W}$ is defined as follows:

$$\mathbf{QDQ}(\mathbf{W}) = s \cdot \text{clip}\Big(\Big\lfloor \frac{\mathbf{W}}{s} \Big\rceil, n, m\Big), \quad n, m \in \mathbb{N}, \tag{1}$$

where the rounding operation $\lfloor \cdot \rceil$ is typically performed using the Round-to-Nearest (RTN) method. The scale factor s is defined as:

$$s = \frac{\max(\mathbf{W}) - \min(\mathbf{W})}{2^{\text{bit}} - 1}, \tag{2}$$

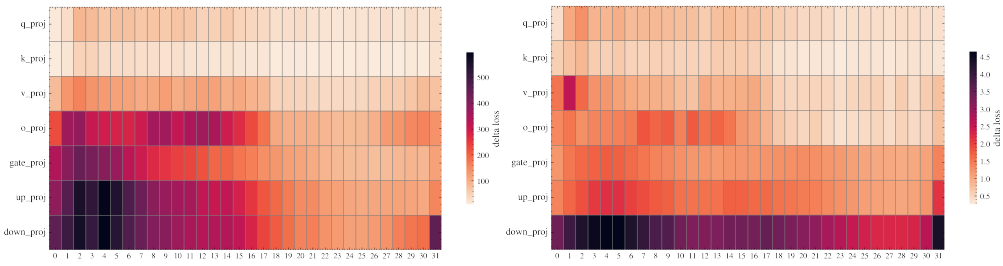
and for simplicity, we ignore the zero point.

SignRoundV1 introduces three trainable parameters v, α and β, to reduce the quantization error. The parameter v enhances the rounding operation

$$\mathbf{QDQ}(\mathbf{W}) = s \cdot \text{clip}\Big(\Big\lfloor \frac{\mathbf{W}}{s} + v \Big\rceil, n, m\Big), \quad n, m \in \mathbb{N}, \tag{3}$$

while α and β refine the scale through

$$s = \frac{\max(\mathbf{W}) \cdot \alpha - \min(\mathbf{W}) \cdot \beta}{2^{\text{bit}} - 1}. \tag{4}$$



(a) W2A16 layer-wise DeltaLoss sensitivity (b) MXFP4 layer-wise DeltaLoss sensitivity

Figure 2: Layer-wise DeltaLoss sensitivity of Llama-3.1-8B-Instruct under W2A16 and MXFP4.

3.1 DELTALOSS SENSITIVITY METRIC

Prior work (Dong et al., 2019; You et al., 2024) largely ignores first-order gradient information and instead relies on second-order approximations, such as the Hessian or Fisher information matrix, to estimate sensitivity under the assumption that gradients are near zero at the local optimum. However, this assumption breaks down when quantization induces a large loss. Moreover, although second-order metrics capture local curvature, they share a fundamental limitation with purely gradient-based methods: both fail to reflect the actual impact of quantization on the loss and therefore provide only a relative trend rather than a reliable measure of the true loss.

To address this limitation, we estimate the loss increase induced by the QDQ operation using a first-order Taylor approximation. Let $\mathcal{L}_{\text{f}}$ and $\mathcal{L}_{\text{q}}$ denote the losses of the full-precision and quantized

models, respectively. We define the quantization-induced loss increase as

$$\Delta\mathcal{L} = \mathcal{L}_{\mathrm{q}} - \mathcal{L}_{\mathrm{f}}. \tag{5}$$

For a layer with full-precision weight tensor W_{f} and activation tensor A_{f}, and their quantized counterparts W_{q} and A_{q}, we approximate the loss increase by a first-order Taylor expansion around the quantized point:

$$\Delta\mathcal{L} \approx g_{aq}^{\top}(A_{\mathrm{q}} - A_{\mathrm{f}}) + g_{wq}^{\top}(W_{\mathrm{q}} - W_{\mathrm{f}}), \tag{6}$$

where the gradients $g_{aq} = \partial\mathcal{L}/\partial A_{\mathrm{q}}$ and $g_{wq} = \partial\mathcal{L}/\partial W_{\mathrm{q}}$ are computed using the quantized model on a small calibration set.

While Eq. 6 provides a signed first-order estimate, positive and negative element-wise contributions may cancel each other out when aggregated over tokens, channels, and calibration samples. We therefore instead use the magnitude of the gradient-weighted perturbation as a more stable sensitivity score:

$$\mathrm{DeltaLoss} = \left\|g_{aq} \odot (A_{\mathrm{q}} - A_{\mathrm{f}})\right\|_1 + \left\|g_{wq} \odot (W_{\mathrm{q}} - W_{\mathrm{f}})\right\|_1, \tag{7}$$

where $\odot$ denotes element-wise multiplication and $\|\cdot\|_1$ denotes summation over all elements. The final layer-wise score is averaged over calibration samples.

This metric combines two complementary factors: the magnitude of the quantization perturbation and the task-aware gradient sensitivity. Layers that exhibit both large quantization deviations and large gradients receive higher DeltaLoss scores and are therefore assigned higher precision under a fixed bit budget.

For joint weight-activation quantization, both weight and activation perturbations can be included as in Eq. 7. In practice, prior studies (Xiao et al., 2023; Wei et al., 2023) indicate that activation quantization often dominates the degradation. Therefore, for memory-efficient mixed weight-activation settings where weights and activations share the same bit-width, we use the activation-induced term as the default sensitivity score:

$$\mathrm{DeltaLoss}^{\mathrm{WA}} = \left\|g_{aq} \odot (A_{\mathrm{q}} - A_{\mathrm{f}})\right\|_1. \tag{8}$$

Figure 2 visualizes the layer-wise sensitivity measured by DeltaLoss. It is evident that sensitivity varies across different data types, making it prohibitively labor-intensive or suboptimal to rely on heuristic rules to manually assign quantization bit-widths for each scheme. Moreover, certain layers within a block, such as `down_proj`, exhibit consistently higher sensitivity, suggesting that traditional block-wise fallback strategies may be suboptimal.

3.2 Layer-wise Bit Allocation

We formulate layer-wise bit allocation as a discrete optimization problem. Let n denote the total number of layers, and let $B = \{b_1, \ldots, b_K\}$ be the set of allowed bit-widths. For each layer $i \in \{1, \ldots, n\}$ and bit-width $b \in B$, we introduce a binary variable $I_{i,b} \in \{0, 1\}$, where $I_{i,b} = 1$ indicates that layer i is assigned bit-width b.

For simplicity, we assume identical bit-widths for both weights and activations, although this formulation can be extended to accommodate different bit-widths or data types across layers.

Let $\Delta L_i(b)$ denote the DeltaLoss when quantizing layer i to b bits. Given a target average bit-width T, the mixed-precision assignment can be written as:

$$\begin{aligned} \min_{I_{i,b}} \quad & \sum_{i=1}^{n} \sum_{b \in B} \Delta L_i(b) I_{i,b} \\ \text{s.t.} \quad & \sum_{b \in B} I_{i,b} = 1, \quad \sum_{i=1}^{n} \sum_{b \in B} b I_{i,b} P_i \le T \sum_{i} P_i, \\ & I_{i,b} \in \{0, 1\}, \quad \forall i = 1, \ldots, n,\ b \in B. \end{aligned} \tag{9}$$

where P_i denotes the number of parameters in layer i. This optimization problem can be solved via dynamic programming (Bellman, 1966) or as an integer linear program (Wolsey, 2020; Bertsimas & Tsitsiklis, 1997), both of which are well-established approaches. We omit the details here and refer the reader to the existing literature.

3.3 TUNING QUANTIZATION PARAMETERS

In SignRoundV1 (Cheng et al., 2024), all quantization parameters are initialized with trivial values, specifically, the weight clipping parameters α and β are set to 1.0, and the rounding perturbation v is initialized to 0. A better initialization could potentially improve performance, similar to the effect of careful initialization in training neural networks (He et al., 2015; Glorot & Bengio, 2010). Although initializing the rounding values is challenging, prior works (Bhalgat et al., 2020; Lin et al., 2024b; Badri & Shaji, 2023; Gerganov, 2023) have proposed training-free methods to search for weight clipping by aligning layer outputs or optimizing certain objectives. While effective, these approaches either incur the high cost of layer-wise forward passes, rely on floating-point zero points for improved accuracy, or overlook the impact of activations.

Inspired primarily by the importance matrix in llama.cpp (Gerganov, 2023), we adopt a simple variant to initialize the scale. The search process is formulated as

$$\min_{s \in S} \frac{1}{N} \sum_{i=1}^{N} \left((W_{\mathrm{f}} - W_{\mathrm{q}}) \odot \overline{A} \right)_i^2, \tag{10}$$

where S denotes a predefined set of candidate scales, and $\overline{A}$ represents the channel-wise average of input activations, calibrated from a set of samples. For symmetric quantization, which is the default setting used in this work, the set $S = \{s_0, \ldots, s_c\}$ is constructed from the following candidates:

$$s_i = \frac{\max(|\mathbf{W}|)}{2^{b-1} + \epsilon_i}, \quad \epsilon_i \in \{-t, -t + \delta, \ldots, t\}. \tag{11}$$

t and δ are fixed for each scheme. Specifically, we set $t = 0.9$ and $\delta = 0.01$ for W2A16. We refine the optimal scale s_{init} with a learnable parameter α, i.e., $s = s_{init} \cdot \alpha$, where $\alpha \in [0, 2.0]$. Finally, we follow SignRoundV1 to tune v in Equation 3. For MXFP4, since the scale is restricted to powers of two, the searching strategy is slightly different.

$$s_i = 2^{\lfloor \log_2(\epsilon_i \max_{w \in \mathbf{W}} |w|) \rfloor}, \quad \epsilon_i \in \{0.5, 1.0, 2.0\}. \tag{12}$$

3.4 LOSS FILTERING

To better balance easy and hard samples, we exclude the top-k largest losses in a batch when computing the mean squared error between the quantized block output and the full-precision block, where k is set to 0.1% of the total number of elements. Let $\mathrm{TopK}(\mathcal{L}, k)$ denote the set of the k largest losses in $\mathcal{L}$. The effective loss used for optimization is defined as follows, where "$\setminus$" denotes the exclusion operation:

$$L_{\text{eff}} = \frac{1}{|\mathcal{L} \setminus \mathrm{TopK}(\mathcal{L}, k)|} \sum_{L_i \in \mathcal{L} \setminus \mathrm{TopK}(\mathcal{L}, k)} L_i. \tag{13}$$

3.5 HYPERPARAMETERS

For DeltaLoss, we use only 16 calibration samples with a sequence length of 256. For tuning, we follow the setup in SignRoundV1. Each transformer block is optimized for 200 steps using signed gradient descent, with a learning rate of $1/\text{steps}$ and a batch size of 8. The sequence length is fixed at 2048. To reduce quantization cost, we lower the default number of calibration samples from 512 to 128. We additionally provide a higher-cost recipe for improved accuracy, denoted as **Ours***, where the number of steps is increased from 200 to 500, the learning rate is set to $2.0/\text{steps}$ and the number of calibration samples is restored to 512. Automatic mixed precision (AMP) is applied throughout to improve tuning efficiency.

Table 1: Average accuracies across five tasks (ARC-Challenge, ARC-Easy, HellaSwag, PIQA, WinoGrande) using W2A16 and mixed W2A16G64 and W4A16G128 at 2.5 bits. Ours* indicates our improved recipe (Section 3.5). "-" denotes missing data from the original work. Detailed data are provided in Appendix H, while quantization cost results are reported in Appendix A.

Avg. Bits	Method	Group Size	Llama2-7B	Llama2-13B	Llama2-70B	Llama3-8B	Llama3-70B
16	16-bit	-	64.66	67.44	72.41	68.64	75.28
2	AQLM	2x8	57.61	62.22	69.85	-	-
	AQLM	1x16	**61.85**	**64.95**	70.84	**64.10**	**70.10**
	QuIP#	-	60.61	64.44	**70.91**	-	-
	EQAT	128	59.50	63.88	68.93	59.37	67.57
	EQAT	64	60.14	63.48	69.48	60.76	67.89
	GPTQ	128	41.56	48.29	34.38	-	-
	AWQ	128	34.74	35.99	35.49	-	-
	OmniQ	128	46.98	53.56	54.87	52.66	60.06
	SRV1	128	54.50	60.72	67.70	55.25	64.76
	Ours	128	57.88	61.88	68.39	57.90	69.02
	Ours*	128	58.67	62.34	68.82	59.97	70.16
	Ours*	64	**59.04**	**62.81**	**69.30**	**60.38**	**70.93**
2.5	Ours	128	59.96	63.65	70.10	62.76	72.37
	Ours*	128	60.28	63.73	70.04	62.60	**72.68**
	Ours	64	60.08	64.09	70.25	**63.67**	71.97
	Ours*	64	**60.44**	**64.57**	**70.60**	63.56	72.32

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate SignRoundV2 on two representative LLM families, LLaMA (Touvron et al., 2023b; Grattafiori et al., 2024) and Qwen (Bai et al., 2023; Yang et al., 2025), under both low-bit and mixed-precision settings. Following SignRoundV1 (Cheng et al., 2024), we use the Pile (Gao et al., 2020) for calibration and adopt symmetric quantization for improved hardware and kernel compatibility. All experiments are conducted on NVIDIA A100 (80GB) GPUs and evaluated using the LM-EVALUATION-HARNESS (Gao et al., 2023). We report results on ARC-Challenge and ARC-Easy (Clark et al., 2018), BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), LAMBADA (Paperno et al., 2016), MMLU (Hendrycks et al., 2020), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), TruthfulQA (Lin et al., 2022), WinoGrande (Sakaguchi et al., 2021), and GSM8K (Cobbe et al., 2021). Unless otherwise specified, bit costs for scales and zero points are omitted; for MXFP4 and MXFP8, we follow the standard definitions in (Alliance, 2023).

4.2 COMPARISON OF WEIGHT-ONLY QUANTIZATION METHODS

Table 1 reports average accuracies across five tasks for W2A16 with and without mixed-precision support. In uniform W2A16 settings, SignRoundV2 consistently outperforms established PTQ baselines, including GPTQ (Frantar et al., 2022), AWQ (Lin et al., 2024b), OmniQuant (Shao et al., 2023), and SignRoundV1, by a substantial margin. Compared to high-complexity methods such as AQLM (Egiazarian et al., 2024), QuIP# (Tseng et al., 2024), and EfficientQAT (Chen et al., 2025), our approach achieves competitive performance on large-scale models while maintaining a significantly lower optimization overhead. Furthermore, at 2.5 average bit-widths, our strategy effectively preserves model quality, demonstrating robust recovery capabilities in extreme compression regimes.

4.3 COMPARISON OF ADAPTIVE BIT-WIDTH QUANTIZATION

Table 2 shows that our method achieves strong accuracy across bit-widths, with particularly large gains in the ultra-low-bit regime. Following SFMP (Nie et al., 2026), we report six-task average accuracy with `acc_norm` preferred, use a group size of 128, account for scale and zero-point overhead, and restrict the bit-width difference to at most 1 bit. We use W2A16G128/W3A16G128 mixtures at 2.5 and 3 bits, and W3A16G128/W4A16G128 mixtures at 3.5 and 4 bits, with uniform 3- and 4-bit results provided as non-mixed baselines. SFMP and AMQ (Lee et al., 2025) use AWQ as their base quantizer. SignRoundV2 outperforms AMQ in all 16 model budget settings and SFMP in 11 of 16, including a 12.24-point recovery-rate gain over AMQ on Llama-3.1-8B at 2.5 bits.

Table 2: Comparison of mixed-bit algorithms. Average accuracy (with acc_norm preferred) and recovery rate across six tasks (see Appendix D for detailed data).

Avg. bits	Method	Llama3.1-8B	Qwen3-8B	Qwen3-14B	Qwen3-32B
16	Ours	75.83	74.07	77.41	78.05
	SFMP	75.01	74.20	77.38	77.99
2.5	AMQ	58.65 (78.19%)	58.62 (79.00%)	66.56 (86.02%)	69.37 (88.95%)
	SFMP	65.97 (87.95%)	**67.26 (90.65%)**	71.69 (92.65%)	73.33 (94.02%)
	Ours	**68.57 (90.43%)**	66.82 (90.20%)	**72.63 (93.83%)**	**74.96 (96.04%)**
3	AMQ	68.78 (91.69%)	68.40 (92.18%)	71.32 (92.17%)	74.02 (94.91%)
	SFMP	69.92 (93.21%)	71.80 (96.77%)	**75.45 (97.51%)**	76.24 (97.76%)
	Ours	**71.64 (94.48%)**	**72.21 (97.49%)**	74.66 (96.45%)	**76.61 (98.16%)**
3.125	W3G128	72.84 (96.06%)	73.24 (98.88%)	76.39 (98.68%)	76.66 (98.22%)
3.5	AMQ	72.56 (96.73%)	71.65 (96.56%)	75.76 (97.91%)	76.01 (97.46%)
	SFMP	73.43 (97.89%)	72.96 (98.33%)	**76.89 (99.37%)**	77.24 (99.04%)
	Ours	**73.94 (97.51%)**	**72.99 (98.54%)**	76.32 (98.59%)	**77.36 (99.12%)**
4	AMQ	73.46 (97.93%)	72.64 (97.90%)	76.62 (99.02%)	77.15 (98.92%)
	SFMP	74.15 (98.85%)	73.54 (99.11%)	**77.13 (99.68%)**	**77.89 (99.87%)**
	Ours	**75.07 (99.00%)**	**73.76 (99.58%)**	77.05 (99.53%)	77.48 (99.27%)
4.125	W4G128	75.06 (98.99%)	74.11 (100.05%)	77.33 (99.90%)	77.75 (99.62%)

We further evaluate our mixed-bit method on harder tasks, including MMLU and GSM8K. As shown in Table 3, our method consistently outperforms AMQ and SFMP across different bit-widths. To further isolate the contribution of DeltaLoss, we align the quantization settings across methods. Specifically, 'DL+AWQ' uses the same AWQ quantizer and asymmetric quantization as SFMP and AMQ, with 128 calibration samples of sequence length 1024. We also remove SFMP's row-column weight reordering and denote the resulting variant as 'SFMP-'. Note that SFMP- still performs mixed-bit allocation at the per-group granularity, whereas DeltaLoss and AMQ use per-tensor allocation. Under these controlled settings, DeltaLoss outperforms AMQ in 21 of 24 scenarios and SFMP- in 15 of 24 scenarios, further demonstrating the effectiveness of our mixed-bit allocation strategy.

Table 3: Comparison of mixed-bit methods on 6-task average accuracy, 5-shot MMLU, and GSM8K. See Appendix D for detailed data.

BPW	Method	Qwen3-8B			Qwen3-14B		
		Avg.6	MMLU	GSM8K	Avg.6	MMLU	GSM8K
16	–	74.07	74.88	87.19	77.41	78.78	88.19
2.5	AMQ	58.62	50.78	15.09	66.56	56.18	48.76
	SFMP-	62.61	56.29	36.69	67.92	65.58	65.88
	DL+AWQ	59.08	55.88	14.03	69.03	67.06	52.69
	SFMP	**67.26**	57.33	49.20	71.69	62.55	59.14
	Ours	66.82	**63.59**	**53.22**	**72.63**	**73.08**	**72.48**
3.0	AMQ	68.40	65.41	73.66	71.32	71.89	79.16
	SFMP-	68.76	67.13	76.57	73.96	73.00	86.20
	DL+AWQ	69.32	69.40	79.30	74.11	75.89	86.66
	SFMP	71.80	66.03	74.34	**75.45**	73.74	85.22
	Ours	**72.21**	**72.03**	**85.44**	74.66	**76.10**	**89.23**

BPW	Method	Qwen3-8B			Qwen3-14B		
		Avg.6	MMLU	GSM8K	Avg.6	MMLU	GSM8K
16	–	74.07	74.88	87.19	77.41	78.78	88.19
3.5	AMQ	71.65	71.71	83.32	75.76	75.87	84.48
	SFMP-	71.38	71.83	84.76	75.71	76.86	**88.48**
	DL+AWQ	72.12	72.47	85.44	74.97	**77.42**	**88.48**
	SFMP	72.96	72.11	84.46	**76.89**	77.00	86.10
	Ours	**72.99**	**73.54**	**87.04**	76.32	77.57	86.43
4.0	AMQ	72.64	73.44	85.75	76.62	76.89	86.58
	SFMP-	73.41	73.44	86.20	76.67	77.76	**88.70**
	DL+AWQ	73.33	73.76	86.73	76.36	77.92	88.25
	SFMP	73.54	73.60	**87.17**	**77.13**	78.14	87.41
	Ours	**73.76**	**73.84**	86.96	77.05	**78.25**	87.57

4.4 Performance under MXFP Quantization

We primarily compare SignRoundV2 with SignRoundV1 and RTN under MXFP (Alliance, 2023) in Table 4. Since the RTN results reported by MatGPTQ (Egiazarian et al., 2025) are not directly aligned with our evaluation settings, we defer the comparison with it to Appendix B. SignRoundV2 improves over RTN in the 4-bit regime and outperforms SignRoundV1 in most scenarios. At average bit-widths of 4.5 and 5 bits, SignRoundV2 approaches full-precision performance on most evaluated models. See Appendix E.2 for detailed MXFP4/8 results, and NVFP4 results are reported in Appendix F. We further investigate the effect of the Hadamard transformation. As shown in Table 5, SignRoundV2 substantially outperforms Hadamard alone, while adding Hadamard to SignRoundV2 (V2+H) yields comparable performance. The combination may offer further benefits on more challenging tasks.

Table 4: Mixed-precision MXFP 4/6-bit quantization results. "DL": DeltaLoss without tuning; "I": Instruct models. See Appendix E.1 for details.

Avg. Bits	Method	Llama3.1-8B-I	Llama3.1-70B-I	Qwen2.5-7B-I	Qwen3-8B	Qwen3-32B
16	16-bit	64.16	70.00	65.67	63.24	67.00
4	RTN	58.31 (90.88%)	68.76 (98.23%)	60.62 (92.32%)	58.54 (92.57%)	64.87 (96.82%)
	SRV1	60.72 (94.64%)	69.01 (98.60%)	**64.06 (97.55%)**	60.25 (95.28%)	**66.92 (99.88%)**
	Ours	**61.47 (95.81%)**	**69.31 (99.01%)**	63.72 (97.03%)	**61.45 (97.17%)**	65.90 (98.36%)
4.5	DL	62.00 (96.63%)	70.17 (100.24%)	64.16 (97.70%)	61.35 (97.01%)	66.26 (98.90%)
	Ours	**63.34 (98.72%)**	**70.31 (100.44%)**	**64.77 (98.63%)**	**62.74 (99.21%)**	**67.10 (100.15%)**
5	DL	63.53 (99.02%)	**70.69 (100.99%)**	65.22 (99.31%)	62.06 (98.13%)	66.41 (99.12%)
	Ours	**64.00 (99.75%)**	70.66 (100.94%)	**65.35 (99.51%)**	**62.81 (99.32%)**	**67.19 (100.28%)**

Table 5: Comparison of SignRound variants and Hadamard transformation at MXFP4. V1, V2, and H denote SignRoundV1, SignRoundV2, and Hadamard transformation, respectively.

Model	Method	ARC-c	ARC-e	BoolQ	Hella	LAMB	MMLU	OBQA	PIQA	Truth.	Wino	GSM8K	AVG.
Qwen3-8B	BF16	55.46	83.38	86.73	57.04	64.00	72.96	31.20	76.61	36.72	68.27	87.57	65.45
	RTN	50.17	76.14	84.98	52.00	58.08	65.90	28.60	73.01	33.66	62.83	81.12	60.59
	V1	51.28	**81.02**	85.20	52.86	58.26	68.51	31.40	75.03	32.44	66.54	77.33	61.81
	H	50.51	79.25	**86.42**	**53.67**	64.43	67.26	29.00	74.48	32.68	66.30	77.18	61.93
	V2	**53.84**	80.98	85.84	53.09	61.91	68.88	**32.00**	75.30	**36.35**	**70.72**	**83.47**	**63.85**
	V2+H	51.96	79.34	85.72	53.49	**65.85**	**69.56**	31.60	**75.52**	35.99	68.27	83.17	63.68
Llama3.1-8B-I	BF16	51.54	81.78	84.04	59.13	73.14	67.96	33.00	80.03	37.21	73.80	75.28	65.17
	RTN	44.37	75.55	80.86	55.19	61.96	57.06	31.80	76.28	29.25	70.80	47.08	57.29
	V1	46.25	79.00	83.09	55.28	69.63	60.55	31.80	77.75	33.41	70.48	61.79	60.82
	H	**48.46**	77.86	83.06	**56.46**	68.25	59.78	32.00	**78.13**	28.76	**73.09**	60.27	60.56
	V2	47.35	**79.63**	82.60	56.10	69.53	**62.00**	**33.60**	77.69	32.80	72.06	61.03	61.31
	V2+H	45.82	79.42	**83.94**	56.01	**69.90**	61.87	31.80	78.02	**33.90**	71.43	**63.38**	**61.41**

4.5 Ablation Studies

We conduct ablation studies on pre-tuning initialization, loss filtering, and the calibration data used by DeltaLoss. As shown in Table 6, loss filtering improves the accuracy on Qwen3 by 0.92 percentage points, while combining initialization and loss filtering improves Llama3.1 by 2.99 points over the V1 baseline. These results highlight the importance of both techniques for ultra-low-bit quantization. We further investigate the effect of calibration data at 2.5 bits using W2A16G128/W3A16G128 mixtures, with scale and zero-point overhead included. Increasing the calibration data improves the average accuracy by 2.06 points on Qwen3 and 0.36 points on Llama3.1. However, the larger calibration set substantially increases VRAM usage and calibration time. We therefore use the smaller setting as the default to provide a better trade-off between accuracy and calibration cost.

Table 6: Ablation studies. Detailed results are reported in Table 28 and Table 29, respectively. `w/o im.` denotes without important matrix.

Ablation Study		
Method	Qwen3-8B	Llama3.1-8B-I
V1	53.59	51.20
V1 + Filtering	**54.51**	52.34
V1 + Init (w/o im.)	54.30	53.34
V1 + Init	54.06	53.85
Ours	54.26	**54.19**

Data Configuration of DeltaLoss		
SeqLen / Samples	Qwen3-8B	Llama3.1-8B-I
256 / 16	54.88	54.87
1024 / 128	**56.94**	**55.23**

5 Conclusion

We present SignRoundV2, a robust post-training quantization framework engineered for extremely low-bit LLM deployment. By integrating a gradient-informed sensitivity metric with a lightweight pre-tuning mechanism, SignRoundV2 ensures stable and accurate quantization under aggressive compression regimes. Extensive experiments demonstrate that our approach maintains high-fidelity performance at 4–5 bits and yields robust results even at 2 bits. These results confirm that effective PTQ can significantly reduce the computational and memory footprints of LLMs, offering a scalable solution for efficient large-scale model inference.

AI USE STATEMENT

In this work, we used generative AI tools to polish the writing, format charts and the manuscript, and organize and verify tabulated data. We did not use generative AI tools for algorithmic innovation, experimental design, execution, or analysis, and the remaining disclosure tasks are not applicable to this work. The authors manually reviewed all AI-assisted revisions, checked the formatted charts and tables against the underlying experimental results, and verified the final manuscript for correctness. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This research aims to advance efficient LLM quantization. SignRoundV2 leverages open-source models and publicly available datasets, adhering to their respective licenses and usage terms. As our method requires only minimal tuning of pre-existing models and is not tied to specific high-risk applications, it carries no significant new ethical risks. We acknowledge the contributions of the creators and maintainers of these resources and provide proper citations to the original sources.

REFERENCES

Open Compute Project Foundation MX Alliance. Ocp microscaling formats (mx) specification version 1.0. open compute project foundation technical specification. https://www.opencompute.org/documents/ocp-microscaling-formats-mx-v1-0-spec-final-pdf., 2023. [Accessed 17-11-2025].

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

Hicham Badri and Appu Shaji. Half-quadratic quantization of large machine learning models, November 2023. URL https://mobiusml.github.io/hqq_blog/.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Richard Bellman. Dynamic programming. *science*, 153(3731):34–37, 1966.

Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2978–2985. IEEE, 2020.

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.

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.

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.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. Efficientqat: Efficient quantization-aware training for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10081–10100, 2025.

Wenhua Cheng, Yiyang Cai, Kaokao Lv, and Haihao Shen. Teq: Trainable equivalent transformation for quantization of llms. *arXiv preprint arXiv:2310.10944*, 2023.

Wenhua Cheng, Weiwei Zhang, Haihao Shen, Yiyang Cai, Xin He, Lv Kaokao, and Yi Liu. Optimize weight rounding via signed gradient descent for the quantization of llms. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 11332–11350, 2024.

Krishna Teja Chitty-Venkata, Jie Ye, and Murali Emani. Mopeq: Mixture of mixed precision quantized experts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4023–4032, 2025.

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.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 293–302, 2019.

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. *arXiv preprint arXiv:2402.10631*, 2024.

Haojie Duanmu, Xiuhong Li, Zhihang Yuan, Size Zheng, Jiangfei Duan, Xingcheng Zhang, and Dahua Lin. Mxmoe: Mixed-precision quantization for moe with accuracy and performance co-design. *arXiv preprint arXiv:2505.05799*, 2025.

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

Vage Egiazarian, Roberto L Castro, Denis Kuznedelev, Andrei Panferov, Eldar Kurtic, Shubhra Pandit, Alexandre Marques, Mark Kurtz, Saleh Ashkboos, Torsten Hoefler, et al. Bridging the gap between promise and performance for microscaling fp4 quantization. *arXiv preprint arXiv:2509.23202*, 2025.

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

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

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

Georgi Gerganov. llama.cpp: LLM inference in C/C++. https://github.com/ggml-org/llama.cpp, 2023.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2020.

Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Improving post training neural quantization: Layer-wise calibration and integer programming. *arXiv preprint arXiv:2006.10518*, 2020.

Damjan Kalajdzievski. Scaling laws for forgetting when fine-tuning large language models. *arXiv preprint arXiv:2401.05605*, 2024.

Wenjing Ke, Zhe Li, Dong Li, Lu Tian, and Emad Barsoum. Dl-qat: Weight-decomposed low-rank quantization-aware training for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 113–119, 2024.

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

Jung Hyun Lee, Jeonghoon Kim, Se Jung Kwon, and Dongsoo Lee. Flexround: Learnable rounding based on element-wise division for post-training quantization. *arXiv preprint arXiv:2306.00317*, 2023.

Sangjun Lee, Seung-taek Woo, Jun-gyu Jin, Changhun Lee, and Eunhyeok Park. Amq: Enabling automl for mixed-precision weight-only quantization of large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 35520–35538, 2025.

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.

Zhengyi Li, Cong Guo, Zhanda Zhu, Yangjie Zhou, Yuxian Qiu, Xiaotian Gao, Jingwen Leng, and Minyi Guo. Efficient activation quantization via adaptive rounding border for post-training quantization. *arXiv preprint arXiv:2208.11945*, 2022.

Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. *Advances in Neural Information Processing Systems*, 37:87766–87800, 2024a.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*, 2024b.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, 2022.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Shih-Yang Liu, Zechun Liu, and Kwang-Ting Cheng. Oscillation-free quantization for low-bit vision transformers. In *International Conference on Machine Learning*, pp. 21813–21824. PMLR, 2023a.

Wenyuan Liu, Haoqian Meng, Yilun Luo, Peng Zhang, and Xindian Ma. Micromix: Efficient mixed-precision quantization with microscaling formats for large language models. *arXiv preprint arXiv:2508.02343*, 2025.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. Llm-qat: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*, 2023b.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations. *arXiv preprint arXiv:2405.16406*, 2024b.

Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems*, 34:28092–28103, 2021.

Qian Lou, Feng Guo, Lantao Liu, Minje Kim, and Lei Jiang. Autoq: Automated kernel-wise neural network quantization. *arXiv preprint arXiv:1902.05690*, 2019.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Lifeng Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 1(4), 2024.

Vladimir Malinovskii, Andrei Panferov, Ivan Ilin, Han Guo, Peter Richtárik, and Dan Alistarh. Pushing the limits of large language model quantization via the linearity theorem. *arXiv preprint arXiv:2411.17525*, 2024.

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.

Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1325–1334, 2019.

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

Xin Nie, Haicheng Zhang, Liang Dong, Beining Feng, Jinhong Weng, and Guiling Sun. Sfmp: Fine-grained, hardware-friendly and search-free mixed-precision quantization for large language models. *arXiv preprint arXiv:2602.01027*, 2026.

NVIDIA. Nvidia blackwell architecture technical brief. `https://resources.nvidia.com/en-us-blackwell-architecture`, 2024. [Accessed 17-11-2025].

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, 2016.

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

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. In *The Twelfth International Conference on Learning Representations*, 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. *arXiv preprint arXiv:2410.09426*, 2024.

Wei Tao, Haocheng Lu, Xiaoyang Qu, Bin Zhang, Kai Lu, Jiguang Wan, and Jianzong Wang. Moqae: Mixed-precision quantization for long-context llm inference via mixture of quantization-aware experts. *arXiv preprint arXiv:2506.07533*, 2025.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

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

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks. *arXiv preprint arXiv:2402.04396*, 2024.

Hongyu Wang, Shuming Ma, Lingxiao Ma, Lei Wang, Wenhui Wang, Li Dong, Shaohan Huang, Huaijie Wang, Jilong Xue, Ruiping Wang, et al. Bitnet: 1-bit pre-training for large language models. *Journal of Machine Learning Research*, 26(125):1–29, 2025.

Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8612–8620, 2019.

Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. Outlier suppression+: Accurate quantization of large language models by equivalent and optimal shifting and scaling. *arXiv preprint arXiv:2304.09145*, 2023.

Laurence A Wolsey. *Integer programming*. John Wiley & Sons, 2020.

Xun Wu, Shaohan Huang, Wenhui Wang, Ting Song, Li Dong, Yan Xia, and Furu Wei. Bitnet distillation. *arXiv preprint arXiv:2510.13998*, 2025.

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

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

Zhewei Yao, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael Mahoney, et al. Hawq-v3: Dyadic neural network quantization. In *International Conference on Machine Learning*, pp. 11875–11886. PMLR, 2021.

Zhewei Yao, Xiaoxia Wu, Cheng Li, Stephen Youn, and Yuxiong He. Exploring post-training quantization in llms from comprehensive study to low rank compensation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19377–19385, 2024.

Haoran You, Yipin Guo, Yichao Fu, Wei Zhou, Huihong Shi, Xiaofan Zhang, Souvik Kundu, Amir Yazdanbakhsh, and Yingyan (Celine) Lin. Shiftaddllm: Accelerating pretrained llms via post-training multiplication-less reparameterization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 24822–24848. Curran Associates, Inc., 2024. doi: 10.52202/079017-0782. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2c30a37c75f062e0bf79297c73db8c6c-Paper-Conference.pdf.

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*. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1472. URL http://dx.doi.org/10.18653/v1/p19-1472.

Chao Zhang, Li Wang, Samson Lasaulce, and Merouane Debbah. Baq: Efficient bit allocation quantization for large language models. *arXiv preprint arXiv:2506.05664*, 2025.

Zhen Zheng, Xiaonan Song, and Chuanjie Liu. Mixllm: Llm quantization with global mixed-precision between output-features and highly-efficient system design. *arXiv preprint arXiv:2412.14590*, 2024.

Yiren Zhou, Seyed-Mohsen Moosavi-Dezfooli, Ngai-Man Cheung, and Pascal Frossard. Adaptive quantization for deep neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

A Quantization Cost

Table 7 compares the end-to-end quantization time cost of several representative methods. Traditional QAT-based and vector quantization approaches such as EfficientQAT (Chen et al., 2025), QuIP# (Tseng et al., 2024), and AQLM (Egiazarian et al., 2024) require tens to hundreds of hours to converge. Besides, we argue that QAT requires careful hyperparameter tuning, and its actual quantization cost is often much higher than what is reported. In contrast, SignRoundV1 completes in only 2.2 hours. Our method maintains a similarly low computational cost, requiring 2.5 hours, while the enhanced version (Ours*) takes 6 hours due to additional tuning samples and steps. Overall, our approach achieves competitive accuracy with substantially lower time overhead compared with prior methods. Table 8 summarizes the VRAM and runtime overhead of computing DeltaLoss for weight-only quantization (e.g., W2A16), with MXFP4 exhibiting comparable cost.

Table 7: Runtime Cost Comparison for Llama2-70B

Method	Fits A100-80GB	GPU hours
SignRoundV1	✓	2.2
EfficientQAT	✓	41
QuIP#	×	270
AQLM	✓	336
Ours	✓	2.5
Ours*	✓	6

Table 8: Cost Comparison for Adaptive Bit-Width Assignment (Excluding Quantization Overhead).

Model	Method	VRAM (GB)	GPU hours
Qwen3-8B	AMQ	20	7
	SFMP	48	0.05
	Ours	15	$0.017\times$len(opt)
Qwen3-32B	AMQ	70	≈ 48
	SFMP	150	≈ 0.2
	Ours	30	$0.05\times$len(opt)
Llama2-70B	AMQ	160	176
	SFMP	350	0.6
	Ours	40	$0.117\times$len(opt)

B Comparison with Other MXFP4 Methods

We have added additional comparison results for GPTQ (Frantar et al., 2022) and MR-GPTQ (Egiazarian et al., 2025). We followed the evaluation protocol described in MR-GPTQ as closely as possible. While the 16-bit results are generally consistent with the reported results in MR-GPTQ, we were unable to reproduce their RTN baseline, so we compare different methods based on their improvements over their respective RTN baselines. As shown in Table 9, our method consistently achieves substantial improvements over the corresponding RTN baseline. Although MR-GPTQ achieves the

highest recovery, it introduces additional Hadamard transforms beyond standard GPTQ and has higher RTN baseline. Under this metric, our method consistently achieves larger gains than GPTQ, highlighting the effectiveness of our method as a lightweight alternative for MXFP4 quantization.

Table 9: MXFP4 quantization results on four reasoning and commonsense benchmarks. Recovery is measured relative to the corresponding full-precision result, and improvement is measured relative to the corresponding RTN result, since the RTN baselines differ across reporters. H denotes Hadamard transformation. **Please note that MR-GPTQ incorporates Hadamard transformation.**

Model	Source	Method	WinoGrande 5-shot	HellaSwag 10-shot	GSM8K 8-shot	MMLU-CoT	Avg. Acc.	Recovery (%)	VS. RTN (%)
Llama-3.1-8B-I		16-bit	0.7790	0.8001	0.8506	0.7276	0.7893	–	–
	MR-GPTQ	RTN	0.7324	0.7399	0.6785	0.6221	0.6932	87.83%	–
		RTN+H	0.7167	0.7529	0.7248	0.6238	0.7046	89.26%	1.63%
		GPTQ	0.7364	0.7552	0.7020	0.6374	0.7078	89.67%	2.10%
		MR-GPTQ	0.7481	0.7691	0.7570	0.6719	0.7365	93.31%	6.25%
	SRV2	16-bit	0.7830	0.8005	0.8529	0.7262	0.7907	–	–
		RTN	0.7017	0.7580	0.6816	0.6244	0.6914	87.45%	–
		Ours	0.7403	0.7521	0.7058	0.6570	0.7138	90.28%	3.24%
Qwen3-8B		16-bit	0.7056	0.7552	0.9090	0.7298	0.7749	–	–
	MR-GPTQ	RTN	0.6740	0.7124	0.8423	0.6769	0.7264	93.74%	–
		RTN+H	0.6732	0.7132	0.8378	0.6757	0.7250	93.56%	-0.20%
		GPTQ	0.6780	0.7165	0.8423	0.6801	0.7292	94.11%	0.39%
		MR-GPTQ	0.6803	0.7317	0.8484	0.6913	0.7379	95.23%	1.59%
	SRV2	16-bit	0.7111	0.7648	0.9105	0.7186	0.7763	–	–
		RTN	0.6677	0.7111	0.7710	0.5673	0.6793	87.51%	–
		Ours	0.6630	0.7130	0.8666	0.6288	0.7179	92.48%	5.68%

C Comparison with Naive Mixed-bit Methods

The results in Tables 10 and 11 demonstrate that our method consistently outperforms simple head-layer and tail-layer heuristics across a range of precision budgets and model architectures. For MXFP quantization (4.5, 5, and 6 bits), the DL strategy achieves the highest average accuracy for all three models without requiring any tuning, showing clear improvements over allocating 4-bit precision only to the head (near the LM head) or tail layers (near the embedding). At lower precision (average 3 bits using W2G128/W4G128), the performance gap becomes even more obvious: both head- and tail-focused heuristics degrade substantially, while our method remains robust and delivers the best accuracy across all evaluated models.

Table 10: Comparison with head-layer (Head) and tail-layer (Tail) heuristics at 4.5, 5, and 6 bits using DeltaLoss without tuning. The average accuracies are computed over 10 tasks; The "I" in the model name is short for "Instruct".

Avg. bits	Method	Llama3.1-8B-I	Qwen2.5-7B-I	Qwen3-8B
4.5	Head 4-bit	58.93	61.38	59.26
	Middle 8-bit	58.96	60.79	59.36
	Middle 4-bit	58.94	62.21	59.50
	Tail 4-bit	59.18	60.63	58.92
	DL	**60.64**	**63.13**	**59.81**
5	Head 4-bit	60.12	61.35	59.65
	Middle 8-bit	59.40	61.03	60.26
	Middle 4-bit	59.61	62.79	60.20
	Tail 4-bit	60.82	61.75	59.43
	DL	**62.57**	**64.04**	**61.45**
6	Head 4-bit	62.65	63.39	61.53
	Middle 8-bit	60.92	62.87	61.42
	Middle 4-bit	61.80	62.91	60.65
	Tail 4-bit	59.88	62.22	60.54
	DL	**63.64**	**65.45**	**62.00**

Table 11: Average accuracies across 10 tasks at an average of 3 bits (W2G128 / W4G128). The "I" in the model name is short for "Instruct".

Avg. bits = 3	Llama3.1-8B-I	Qwen2.5-7B-I	Qwen3-8B
Head 4-bit	31.98	32.70	31.96
Tail 4-bit	60.58	37.98	45.36
Ours	**61.48**	**40.58**	**48.62**

D Comparison with Prior Mixed-bit Methods

This appendix provides the detailed six-task accuracies for the mixed-bit weight-only quantization results reported in Section 4.3. We report HellaSwag, WinoGrande, ARC-Easy, ARC-Challenge, PIQA, BoolQ, and the average accuracy (AVG) for each model and method. Specifically, Table 12 reports results for Llama-3.1-8B, Table 13 for Qwen3-8B, Table 14 for Qwen3-14B, and Table 15 for Qwen3-32B.

Table 12: Detailed accuracies on six tasks for **Llama3.1-8B** with mixed-bit weight-only quantization. "Avg. Bits" indicates the average bit-width.

Avg. bits	Method	ARC-c	ARC-e	BoolQ	Hella.	PIQA	Wino.	AVG
16	Ours	54.78	82.70	83.12	79.26	80.90	74.19	75.83
	SFMP-16-bit	53.41	81.19	82.15	78.99	81.39	72.93	75.01
2.5	AMQ	34.89	59.63	65.57	57.18	71.00	63.61	58.65
	SFMP	41.21	68.64	77.13	65.37	74.59	68.90	65.97
	Ours	**45.90**	**73.40**	**78.38**	**69.02**	**75.57**	**69.14**	**68.57**
3	AMQ	45.48	72.69	76.48	70.38	77.64	70.01	68.78
	SFMP	44.11	73.40	77.92	72.43	**79.33**	**72.38**	69.92
	Ours	**49.06**	**76.60**	**79.88**	**74.75**	78.18	71.35	**71.64**
3.125	W3G128	51.88	78.79	81.44	75.87	78.18	70.88	72.84
3.5	AMQ	49.57	77.10	80.00	76.15	79.54	73.01	72.56
	SFMP	50.34	77.53	**81.35**	76.61	**80.47**	**74.30**	73.43
	Ours	**52.82**	**80.18**	**81.35**	**76.94**	79.16	73.16	**73.94**
4	AMQ	50.68	78.20	81.04	77.83	79.92	73.09	73.46
	SFMP	51.71	78.58	**82.09**	78.02	**81.23**	73.32	74.15
	Ours	**54.18**	**81.82**	81.93	**78.38**	79.60	**74.51**	**75.07**
4.125	W4G128	54.01	81.65	82.42	78.84	80.03	73.40	75.06

E Mixed MXFP Results

E.1 Mixed MXFP4/6 Results

This subsection reports the detailed results for mixed MXFP4/6 quantization. We list the task-wise accuracies across 10 tasks for each model provided in Tables 16, 17, 18, 19, and 20.

E.2 Mixed MXFP4/8 Results

This subsection reports the detailed results for mixed MXFP4/8 quantization. Table 21 reports the average accuracies and recovery rates across 10 tasks, for MXFP4/8 mixed-precision quantization. The results show that our method generally improves over RTN and achieves competitive or better accuracy than SignRoundV1 (SRV1) in the 4-bit setting. At higher average bit widths (4.5–6 bits), DeltaLoss-only (DL) and our method approach full-precision accuracy, with our method achieving the best recovery in most scenarios. Overall, these results demonstrate that our approach effectively preserves model performance under extreme low-bit mixed-precision quantization while minimizing accuracy loss. The corresponding task-wise accuracies are provided in Tables 22, 23, 24, 25, and 26.

Table 13: Detailed accuracies on six tasks for **Qwen3-8B** with mixed-bit weight-only quantization. "Avg. Bits" indicates the average bit-width. DL+AWQ applies DeltaLoss allocation with the aligned AWQ quantizer, while SFMP- removes SFMP's row–column weight reordering.

Avg. bits	Method	ARC-c	ARC-e	BoolQ	Hella.	PIQA	Wino.	AVG
16	Ours	56.14	80.77	86.61	74.98	77.97	67.96	74.07
	SFMP-16-bit	56.65	80.85	86.64	74.93	77.47	68.66	74.20
2.5	AMQ	35.75	56.19	75.90	55.76	69.70	58.41	58.62
	SFMP-	40.02	67.76	80.46	58.05	70.46	58.88	62.61
	DL+AWQ	33.87	54.80	78.81	56.33	71.06	59.59	59.08
	SFMP	**46.16**	72.05	**83.70**	**62.61**	**73.94**	65.11	**67.26**
	Ours	44.97	**72.81**	81.77	62.30	73.29	**65.75**	66.82
3	AMQ	47.61	73.78	84.40	66.71	73.94	63.93	68.40
	SFMP-	47.70	72.64	85.26	68.29	75.08	63.61	68.76
	DL+AWQ	47.95	75.72	84.04	67.86	75.68	64.64	69.32
	SFMP	53.07	77.61	85.17	**70.52**	76.33	68.11	71.80
	Ours	**53.58**	**78.70**	**86.12**	68.74	**76.61**	**69.53**	**72.21**
3.125	W3G128	55.55	80.30	85.72	71.50	77.69	68.67	73.24
3.5	AMQ	51.02	77.06	**86.40**	71.42	76.93	67.08	71.65
	SFMP-	51.19	76.85	85.29	72.27	75.68	67.01	71.38
	DL+AWQ	52.47	77.27	85.78	72.16	76.44	**68.59**	72.12
	SFMP	**55.03**	78.41	86.06	**73.21**	76.55	68.51	72.96
	Ours	54.10	**80.05**	86.39	71.93	**77.26**	68.19	**72.99**
4	AMQ	53.92	78.49	85.29	73.64	77.25	67.27	72.64
	SFMP-	54.78	79.00	85.69	73.47	**77.91**	**69.61**	73.41
	DL+AWQ	**56.31**	78.91	**86.76**	73.53	77.20	67.25	73.33
	SFMP	55.12	79.38	85.88	**74.63**	77.09	69.14	73.54
	Ours	**56.31**	**79.67**	86.61	73.77	77.53	68.67	**73.76**
4.125	W4G128	57.42	80.26	86.88	73.79	77.26	69.06	74.11

Table 14: Detailed accuracies on six tasks for **Qwen3-14B** with mixed-bit weight-only quantization. "Avg. Bits" indicates the average bit-width. DL+AWQ applies DeltaLoss allocation with the aligned AWQ quantizer, while SFMP- removes SFMP's row–column weight reordering.

Avg. bits	Method	ARC-c	ARC-e	BoolQ	Hella.	PIQA	Wino.	AVG
16	Ours	60.49	82.74	89.39	78.80	80.03	73.01	77.41
	SFMP-16-bit	60.41	82.79	89.33	78.92	79.98	72.84	77.38
2.5	AMQ	44.18	70.47	84.39	64.31	72.09	63.90	66.56
	DL+AWQ	46.33	73.11	85.93	66.41	75.57	66.85	69.03
	SFMP	50.85	76.22	86.94	69.61	76.93	69.30	71.69
	Ours	**51.62**	**78.58**	**87.98**	**70.21**	**77.09**	**70.32**	**72.63**
3	AMQ	50.27	75.89	85.33	71.16	75.94	69.34	71.32
	DL+AWQ	53.84	78.91	**88.72**	74.25	78.13	70.80	74.11
	SFMP	**57.59**	**80.68**	88.17	**75.80**	**78.40**	**72.06**	**75.45**
	Ours	57.08	80.26	87.37	74.15	77.58	71.51	74.66
3.125	W3G128	59.56	82.15	88.01	76.12	79.27	73.24	76.39
3.5	AMQ	58.31	81.56	87.56	76.04	79.12	71.98	75.76
	DL+AWQ	56.48	79.97	88.44	76.95	78.67	69.30	74.97
	SFMP	**59.98**	**82.49**	88.96	**77.64**	**79.60**	**72.93**	**76.89**
	Ours	59.04	81.94	**88.99**	77.05	79.22	71.67	76.32
4	AMQ	59.42	82.05	88.76	77.68	79.65	72.13	76.62
	DL+AWQ	59.13	82.20	88.35	78.09	78.62	71.74	76.36
	SFMP	**60.41**	**83.08**	**89.02**	**78.23**	79.60	72.45	**77.13**
	Ours	59.81	82.58	88.90	78.19	**79.71**	**73.09**	77.05
4.125	W4G128	61.01	82.91	89.08	78.17	80.09	72.69	77.33

Table 15: Detailed accuracies on six tasks for **Qwen3-32B** with mixed-bit weight-only quantization. "Avg. Bits" indicates the average bit-width.

Avg. bits	Method	ARC-c	ARC-e	BoolQ	Hella.	PIQA	Wino.	AVG
16	Ours	60.92	83.29	86.42	82.60	81.99	73.09	78.05
	SFMP-16-bit	60.92	83.25	86.42	82.56	81.88	72.93	77.99
2.5	AMQ	50.63	73.90	80.08	71.68	75.74	64.19	69.37
	SFMP	55.89	78.41	82.26	**76.93**	**79.16**	67.32	73.33
	Ours	**55.97**	**79.21**	**88.41**	74.50	78.24	**73.40**	**74.96**
3	AMQ	58.83	79.62	83.26	77.10	77.14	68.15	74.02
	SFMP	**59.64**	**81.35**	86.70	**80.00**	79.27	70.48	76.24
	Ours	58.79	80.81	**89.02**	78.29	**79.49**	**73.24**	**76.61**
3.125	W3G128	60.32	82.74	85.20	78.66	79.92	73.09	76.66
3.5	AMQ	59.71	81.15	84.78	80.02	79.14	71.26	76.01
	SFMP	60.41	82.02	**85.88**	**81.18**	**81.42**	72.53	77.24
	Ours	**62.37**	**82.91**	85.14	80.57	80.58	**72.61**	**77.36**
4	AMQ	60.87	82.31	85.42	81.58	80.95	71.76	77.15
	SFMP	**61.09**	**83.46**	**86.20**	**82.01**	**81.73**	72.83	**77.89**
	Ours	61.01	82.11	85.05	80.98	81.39	**74.35**	77.48
4.125	W4G128	61.43	82.37	85.75	82.24	81.28	73.40	77.75

Table 16: The detailed accuracies of 10 tasks for **Llama-3.1-8B-Instruct** model with MXFP4/6 mixed precision. "Avg. Bits" indicates average per-weight bit-width. "DL": DeltaLoss without tuning.

Avg. Bits	Method	ARC-c.	ARC-e	BoolQ	Hella.	Lamb.	MMLU	Open.	PIQA	Truth.	Wino.	Avg.
16	16-bit	51.54	81.78	84.04	59.13	73.14	67.96	33.00	80.03	37.21	73.80	64.16
4	RTN	44.37	75.55	80.86	55.19	61.96	57.06	31.80	76.28	29.25	70.80	58.31
	SRV1	46.25	79.00	83.09	55.28	**69.63**	60.55	31.80	77.75	**33.41**	70.48	60.72
	Ours	**50.68**	**80.05**	**83.12**	**56.28**	69.05	**61.69**	**32.40**	**78.13**	32.19	**71.11**	**61.47**
4.5	DL	48.38	79.29	83.88	**58.31**	68.48	64.51	31.60	78.78	35.01	71.74	62.00
	Ours	**50.43**	**80.13**	**84.86**	57.83	**71.36**	**65.75**	**34.60**	**79.43**	**36.60**	**72.45**	**63.34**
5	DL	50.94	81.06	**85.11**	**58.90**	68.64	66.47	**35.00**	**79.54**	35.25	74.35	63.53
	Ours	**51.11**	**81.52**	84.68	58.77	**71.76**	**66.81**	34.60	79.43	**36.35**	**74.98**	**64.00**

Table 17: The detailed accuracies of 10 tasks for **Llama-3.1-70B-Instruct** model with MXFP4/6 mixed precision. "Avg. Bits" indicates average per-weight bit-width. "DL": DeltaLoss without tuning.

Avg. Bits	Method	ARC-c.	ARC-e	BoolQ	Hella.	Lamb.	MMLU	Open.	PIQA	Truth.	Wino.	Avg.
16	16-bit	62.46	86.87	87.80	65.15	75.51	82.34	37.20	83.30	40.64	78.69	70.00
4	RTN	**59.56**	**85.10**	88.81	63.16	73.36	79.03	36.20	82.10	**40.88**	79.40	68.76
	SRV1	58.28	85.06	88.53	**63.35**	**75.24**	79.66	**37.60**	**82.32**	40.15	**79.95**	69.01
	Ours	59.39	85.02	**89.27**	63.12	74.75	**79.92**	**37.60**	82.21	40.64	81.14	**69.31**
4.5	DL	**61.01**	**86.32**	88.96	65.09	75.53	81.01	**37.40**	**83.03**	**41.86**	81.53	70.17
	Ours	59.98	85.94	**89.24**	**65.10**	**76.98**	**81.21**	36.80	**83.03**	41.00	**83.82**	**70.31**
5	DL	**61.86**	**86.57**	**88.81**	65.57	76.79	81.75	37.40	**83.35**	**42.59**	82.16	**70.69**
	Ours	61.35	86.36	88.75	**65.89**	**76.98**	**82.19**	**38.00**	**83.35**	40.76	**82.95**	70.66

Table 18: The detailed accuracies of 10 tasks for **Qwen2.5-7B-Instruct** model with MXFP4/6 mixed precision. "Avg. Bits" indicates average per-weight bit-width. "DL": DeltaLoss without tuning.

Avg. Bits	Method	ARC-c.	ARC-e	BoolQ	Hella.	Lamb.	MMLU	Open.	PIQA	Truth.	Wino.	Avg.
16	16-bit	52.56	81.61	86.36	62.01	69.44	71.76	34.60	79.65	47.98	70.72	65.67
4	RTN	48.81	75.59	82.48	56.84	57.68	65.15	34.80	75.79	41.62	67.48	60.62
	SRV1	**53.24**	**80.89**	85.47	57.88	64.41	68.05	**35.00**	77.48	**46.76**	**71.43**	**64.06**
	Ours	51.28	80.81	**85.72**	**58.16**	**66.47**	**68.49**	34.20	**77.64**	45.29	69.14	63.72
4.5	DL	50.60	80.13	**86.51**	59.27	**69.22**	69.73	33.80	76.66	**45.53**	**70.17**	64.16
	Ours	**53.92**	**81.73**	86.21	**59.56**	69.03	**70.43**	**34.60**	**77.20**	45.41	69.61	**64.77**
5	DL	52.47	81.78	**86.73**	60.66	68.41	70.32	**36.80**	77.69	46.39	**70.96**	65.22
	Ours	**53.50**	**82.07**	86.33	**60.75**	**69.36**	**70.77**	35.00	**78.07**	**46.88**	70.80	**65.35**

Table 19: The detailed accuracies of 10 tasks for **Qwen3-8B** model with MXFP4/6 mixed precision. "Avg. Bits" indicates average per-weight bit-width. "DL": DeltaLoss without tuning.

Avg. Bits	Method	ARC-c.	ARC-e	BoolQ	Hella.	Lamb.	MMLU	Open.	PIQA	Truth.	Wino.	Avg.
16	16-bit	55.46	83.38	86.73	57.04	64.00	72.96	31.20	76.61	36.72	68.27	63.24
4	RTN	50.17	76.14	84.98	52.00	58.08	65.90	28.60	73.01	33.66	62.83	58.54
	SRV1	51.28	**81.02**	85.20	**52.86**	58.26	68.51	**31.40**	75.03	32.44	66.54	60.25
	Ours	**53.92**	81.65	**85.50**	52.75	**62.14**	**69.41**	30.40	**75.19**	**35.62**	**67.88**	**61.45**
4.5	DL	52.05	80.30	85.96	53.96	62.57	70.99	**31.00**	74.32	34.27	68.03	61.35
	Ours	**54.95**	**82.66**	**86.67**	**54.82**	**64.37**	**71.52**	30.80	**76.12**	**37.33**	**68.19**	**62.74**
5	DL	51.54	80.22	**87.25**	55.43	62.41	**71.96**	**31.80**	76.17	**35.62**	68.19	62.06
	Ours	**54.44**	**82.53**	86.97	**55.75**	**64.27**	71.90	**31.80**	**76.33**	**35.62**	**68.51**	**62.81**

Table 20: The detailed accuracies of 10 tasks for **Qwen3-32B** model with MXFP4/6 mixed precision. "Avg. Bits" indicates average per-weight bit-width. "DL": DeltaLoss without tuning.

Avg. Bits	Method	ARC-c.	ARC-e	BoolQ	Hella.	Lamb.	MMLU	Open.	PIQA	Truth.	Wino.	Avg.
16	16-bit	57.94	84.47	86.39	63.91	67.13	80.74	36.00	80.96	39.05	73.40	67.00
4	RTN	56.66	81.23	84.95	**61.73**	65.96	76.59	33.80	77.97	39.17	70.64	64.87
	SRV1	**57.51**	**84.55**	**87.13**	61.61	**68.83**	78.32	**36.20**	**79.71**	**41.37**	**73.95**	**66.92**
	Ours	55.97	82.70	85.87	61.23	68.79	**78.41**	34.60	78.62	40.15	72.69	65.90
4.5	DL	**57.08**	83.25	86.39	63.07	66.08	79.48	33.40	**80.96**	**42.11**	70.80	66.26
	Ours	56.57	**84.22**	**88.04**	**63.44**	**68.33**	**79.91**	**36.00**	80.14	40.88	**73.48**	**67.10**
5	DL	57.85	**84.22**	85.20	63.35	67.53	79.84	**35.60**	79.27	**39.53**	71.67	66.41
	Ours	**60.15**	**84.22**	**87.25**	**63.38**	**68.45**	**80.15**	**35.40**	**80.09**	**39.53**	**73.32**	**67.19**

Table 21: Average accuracies and recovery rates (%) across 10 tasks, with MXFP4 at 4 bits and mixed MXFP4/8 for the higher-bit settings. "Avg. Bits" indicates the average per-weight bit-width. "DL" denotes DeltaLoss without tuning. The "I" in the model name is short for "Instruct".

Avg. Bits	Method	Llama3.1-8B-I	Llama3.1-70B-I	Qwen2.5-7B-I	Qwen3-8B	Qwen3-32B
16	16-bit	64.16	70.00	65.67	63.24	67.00
4	RTN	58.31 (90.88%)	68.71 (98.16%)	60.62 (92.32%)	58.54 (92.57%)	65.07 (97.12%)
	SRV1	60.72 (94.64%)	69.01 (98.60%)	**64.06 (97.55%)**	60.25 (95.28%)	**66.92 (99.88%)**
	Ours	**61.34 (95.59%)**	**69.32 (99.04%)**	63.37 (96.50%)	**61.89 (97.87%)**	66.86 (99.79%)
4.5	DL	60.64 (94.50%)	69.78 (99.70%)	63.13 (96.13%)	59.81 (94.58%)	66.05 (98.58%)
	Ours	**62.33 (97.14%)**	**69.96 (99.95%)**	**64.51 (98.24%)**	**62.28 (98.49%)**	**66.89 (99.83%)**
5	DL	62.57 (97.51%)	70.10 (100.15%)	64.04 (97.52%)	61.45 (97.18%)	66.00 (98.51%)
	Ours	**63.19 (98.49%)**	**70.31 (100.45%)**	**65.04 (99.03%)**	**62.54 (98.89%)**	**67.17 (100.25%)**
6	DL	63.64 (99.18%)	**70.56 (100.80%)**	**65.45 (99.67%)**	62.00 (98.04%)	66.61 (99.42%)
	Ours	**64.12 (99.93%)**	70.50 (100.71%)	65.30 (99.43%)	**62.19 (98.34%)**	**67.21 (100.32%)**

Table 22: The detailed accuracies of 10 tasks for **Llama-3.1-8B-Instruct** model with MXFP4/8 mixed precision. "Avg. Bits" indicates average per-weight bit-width. "DL" denotes DeltaLoss-only without tuning.

Avg. Bits	Method	ARC-c.	ARC-e	BoolQ	Hella.	Lamb.	MMLU	Open.	PIQA	Truth.	Wino.	Avg.
16	16-bit	51.54	81.78	84.04	59.13	73.14	67.96	33.00	80.03	37.21	73.80	64.16
4	RTN	44.37	75.55	80.86	55.19	61.96	57.06	31.80	76.28	29.25	70.80	58.31
	SRV1	46.25	79.00	**83.09**	55.28	**69.63**	60.55	31.80	**77.75**	**33.41**	70.48	60.72
	Ours	**47.35**	**79.63**	82.60	**56.10**	69.53	**62.00**	**33.60**	77.69	32.80	**72.06**	**61.34**
4.5	DL	47.18	77.61	**83.70**	**57.57**	63.67	62.83	**33.60**	77.64	31.21	**71.35**	60.64
	Ours	**49.06**	**80.43**	83.18	57.35	**71.41**	**64.86**	33.40	**78.40**	**34.76**	70.40	**62.33**
5	DL	**50.43**	79.17	84.10	**58.22**	68.62	64.77	34.60	78.62	34.52	**72.61**	62.57
	Ours	49.83	**80.93**	**84.19**	57.92	**72.19**	**65.43**	**35.80**	**79.22**	**35.37**	71.03	**63.19**
6	DL	51.45	81.19	84.59	**58.94**	69.73	65.98	**36.40**	79.33	34.76	74.03	63.64
	Ours	**51.54**	**81.40**	**84.89**	58.57	**71.96**	**66.85**	35.40	**79.38**	**36.84**	**74.35**	**64.12**

Table 23: The detailed accuracies of 10 tasks for **Llama-3.1-70B-Instruct** model with MXFP4/8 mixed precision. "Avg. Bits" indicates average per-weight bit-width. "DL" denotes DeltaLoss-only without tuning.

Avg. Bits	Method	ARC-c.	ARC-e	BoolQ	Hella.	Lamb.	MMLU	Open.	PIQA	Truth.	Wino.	Avg.
16	16-bit	62.46	86.87	87.80	65.15	75.51	82.34	37.20	83.30	40.64	78.69	70.00
4	RTN	59.47	85.10	**88.90**	63.18	73.53	78.72	36.00	**82.32**	**40.88**	79.01	68.71
	SRV1	58.28	85.06	88.53	63.35	75.24	79.66	**37.60**	**82.32**	40.15	79.95	69.01
	Ours	**59.90**	**85.82**	88.53	**63.63**	**75.45**	**79.97**	37.00	81.88	40.76	**80.27**	**69.32**
4.5	DL	**59.73**	**86.24**	**88.69**	**64.68**	75.28	80.70	36.40	82.10	**41.62**	**82.40**	69.78
	Ours	59.64	**86.24**	**88.69**	64.51	**76.89**	**81.20**	**37.20**	**82.86**	40.39	82.00	**69.96**
5	DL	60.15	86.57	88.84	**64.94**	75.65	81.08	**38.20**	82.70	**41.13**	**81.77**	70.10
	Ours	**61.18**	**86.95**	**89.17**	64.89	**77.18**	**81.23**	37.40	**83.19**	41.00	80.90	**70.31**
6	DL	61.09	**86.53**	**88.93**	65.45	**76.89**	81.99	**38.00**	83.08	**41.13**	**82.48**	**70.56**
	Ours	**62.12**	86.15	88.72	**65.65**	76.69	**82.00**	37.60	**83.30**	40.88	81.85	70.50

Table 24: The detailed accuracies of 10 tasks for **Qwen2.5-7B-Instruct** model with MXFP4/8 mixed precision. "Avg. Bits" indicates average per-weight bit-width. "DL" denotes DeltaLoss-only without tuning.

Avg. Bits	Method	ARC-c.	ARC-e	BoolQ	Hella.	Lamb.	MMLU	Open.	PIQA	Truth.	Wino.	Avg.
16	16-bit	52.56	81.61	86.36	62.01	69.44	71.76	34.60	79.65	47.98	70.72	65.67
4	RTN	48.81	75.59	82.48	56.84	57.68	65.15	34.80	75.79	41.62	67.48	60.62
	SRV1	53.24	**80.89**	**85.47**	57.88	64.41	68.05	**35.00**	**77.48**	**46.76**	**71.43**	**64.06**
	Ours	**53.67**	80.18	85.08	**57.99**	**66.27**	**68.30**	33.60	76.66	43.08	68.90	63.37
4.5	DL	50.09	77.74	**85.96**	58.98	67.86	68.39	**34.60**	75.84	43.33	68.51	63.13
	Ours	**53.50**	**81.65**	85.90	**59.32**	**69.05**	**69.59**	33.60	**78.62**	**44.92**	**68.98**	**64.51**
5	DL	51.11	80.26	**86.85**	59.54	68.27	69.42	**35.20**	76.77	44.31	68.67	64.04
	Ours	**54.61**	**82.62**	85.90	**59.78**	**69.67**	**70.43**	33.60	**78.07**	**45.90**	**69.77**	**65.04**
6	DL	53.50	**82.45**	**86.51**	**60.89**	68.93	**70.64**	**36.80**	77.91	**46.39**	**70.48**	**65.45**
	Ours	**54.18**	82.24	86.30	60.68	**69.73**	70.52	34.80	**78.73**	45.78	70.01	65.30

Table 25: The detailed accuracies of 10 tasks for **Qwen3-8B** model with MXFP4/8 mixed precision. "Avg. Bits" indicates average per-weight bit-width. "DL" denotes DeltaLoss-only without tuning.

Avg. Bits	Method	ARC-c.	ARC-e	BoolQ	Hella.	Lamb.	MMLU	Open.	PIQA	Truth.	Wino.	Avg.
16	16-bit	55.46	83.38	86.73	57.04	64.00	72.96	31.20	76.61	36.72	68.27	63.24
4	RTN	50.17	76.14	84.98	52.00	58.08	65.90	28.60	73.01	33.66	62.83	58.54
	SRV1	51.28	**81.02**	85.20	52.86	58.26	68.51	31.40	75.03	32.44	66.54	60.25
	Ours	**53.84**	80.98	**85.84**	**53.09**	**61.91**	**68.88**	**32.00**	**75.30**	**36.35**	**70.72**	**61.89**
4.5	DL	50.51	79.38	86.02	52.96	60.76	68.91	28.60	73.78	33.17	64.01	59.81
	Ours	**53.75**	**81.61**	**86.21**	**54.26**	**64.16**	**70.25**	**31.40**	**75.57**	**37.09**	**68.51**	**62.28**
5	DL	51.11	80.30	86.09	54.23	62.04	70.26	**31.40**	75.19	35.62	**68.27**	61.45
	Ours	**54.10**	**81.82**	**86.42**	**54.62**	**64.20**	**71.53**	30.80	**76.01**	**37.58**	**68.27**	**62.54**
6	DL	**53.84**	81.57	86.94	55.36	62.10	71.46	**31.20**	75.95	35.50	66.06	62.00
	Ours	52.65	**81.65**	**86.97**	**55.57**	**64.06**	**71.54**	30.20	**76.17**	**35.74**	**67.32**	**62.19**

Table 26: The detailed accuracies of 10 tasks for **Qwen3-32B** model with MXFP4/8 mixed precision. "Avg. Bits" indicates average per-weight bit-width. "DL" denotes DeltaLoss-only without tuning.

Avg. Bits	Method	ARC-c.	ARC-e	BoolQ	Hella.	Lamb.	MMLU	Open.	PIQA	Truth.	Wino.	Avg.
16	16-bit	57.94	84.47	86.39	63.91	67.13	80.74	36.00	80.96	39.05	73.40	67.00
4	RTN	55.38	81.82	85.54	61.54	65.67	76.39	34.40	78.94	38.80	72.22	65.07
	SRV1	57.51	**84.55**	87.13	61.61	**68.83**	78.32	**36.20**	**79.71**	41.37	**73.95**	**66.92**
	Ours	**59.47**	83.16	**87.19**	**61.70**	68.78	**78.39**	35.40	**79.71**	**41.74**	73.01	66.86
4.5	DL	56.23	83.33	86.88	62.40	67.49	78.93	**34.60**	79.11	40.64	70.88	66.05
	Ours	**57.59**	**84.09**	**88.23**	**62.51**	**69.42**	**79.51**	33.00	**79.49**	**41.00**	**74.03**	**66.89**
5	DL	56.83	83.38	84.50	63.18	66.25	79.58	34.00	79.92	40.39	71.98	66.00
	Ours	**57.85**	**84.47**	**87.92**	**63.33**	**67.22**	**80.10**	**35.60**	**80.25**	**40.51**	**74.43**	**67.17**
6	DL	57.51	83.04	86.12	63.28	67.69	80.02	**36.40**	80.36	40.15	71.51	66.61
	Ours	**58.19**	**83.88**	**86.85**	**63.44**	**68.12**	**80.40**	**36.40**	**80.47**	**40.88**	**73.48**	**67.21**

F NVFP4 RESULTS

This section provides detailed NVFP4 accuracy results reported in Table 27. Our method achieves the best average accuracy on four out of five models and remains competitive on Qwen3-32B.

Table 27: Comparison of SignRoundV1 (SRV1), RTN, and our method under NVFP4 quantization. We report accuracy (%) across 10 tasks (listed in Section 4.1) and the average (AVG). The "I" in the model name is short for "Instruct".

Model	Method	ARC-c.	ARC-e	BoolQ	Hella.	Lamb.	MMLU	Open.	PIQA	Truth.	Wino.	AVG
Llama3.1-8B-I	16-bit	53.75	82.15	85.35	59.79	72.06	68.28	35.60	80.30	38.07	73.32	64.87
	RTN	50.60	80.01	83.43	**58.30**	71.34	64.13	**35.00**	78.89	34.27	71.90	62.79
	SRV1	50.26	80.39	84.28	58.23	71.07	**65.39**	34.00	79.22	35.86	72.77	63.15
	Ours	**50.68**	**81.65**	**85.44**	57.93	**71.49**	64.91	**35.00**	**79.49**	**37.58**	**74.51**	**63.87**
Llama3.1-70B-I	16-bit	61.09	86.20	89.11	66.28	76.46	82.52	38.80	83.73	41.98	83.43	70.96
	RTN	60.58	85.94	**89.45**	65.03	75.47	80.54	37.20	**83.24**	**42.11**	81.14	70.07
	SRV1	59.73	85.98	88.84	64.77	75.59	**81.31**	**39.00**	83.13	40.88	**82.32**	70.16
	Ours	**60.75**	**86.87**	88.87	**65.10**	**75.82**	81.25	38.20	82.54	41.00	81.29	**70.17**
Qwen2.5-7B-I	16-bit	53.07	81.69	86.33	62.06	69.38	71.71	34.60	79.71	47.98	71.03	65.76
	RTN	51.96	80.18	86.24	60.65	67.15	70.47	33.20	78.51	43.82	67.96	64.01
	SRV1	52.82	81.14	86.09	**60.90**	69.07	70.61	**34.80**	**78.67**	**47.61**	70.40	65.21
	Ours	**54.18**	**81.69**	**86.79**	60.23	**69.65**	70.67	34.00	78.51	47.12	**70.96**	**65.38**
Qwen3-8B	16-bit	55.20	83.54	86.61	57.16	64.10	72.90	31.60	76.66	36.47	67.96	63.22
	RTN	53.41	82.37	85.87	55.29	62.45	70.82	30.80	**76.44**	33.66	67.56	61.87
	SRV1	54.86	82.91	**86.88**	55.31	62.95	**71.97**	**31.20**	76.28	36.35	67.48	62.62
	Ours	**55.29**	**83.04**	86.85	**55.41**	**63.71**	71.36	29.20	75.79	**37.94**	**69.22**	**62.78**
Qwen3-32B	16-bit	58.02	84.39	86.42	63.89	67.13	80.74	36.20	81.23	38.56	73.09	66.97
	RTN	57.51	84.01	87.03	**63.39**	65.83	**79.80**	35.40	79.43	39.53	71.51	66.34
	SRV1	**59.56**	84.22	**88.01**	62.99	68.29	79.50	**37.60**	79.54	**41.00**	72.22	**67.29**
	Ours	57.51	**84.68**	86.51	63.09	**68.89**	**79.80**	36.00	**79.60**	39.05	**72.38**	66.75

G ABLATION STUDY DETAILED DATA

Table 28: Ablation studies of loss filtering and pre-tuning initialization at W2A16G64.

Model	Method	ARC-c	ARC-e	BoolQ	Hella	LAMB	MMLU	OBQA	PIQA	Truth.	Wino	AVG.
Qwen3-8B	V1	38.65	73.11	**80.83**	44.60	48.28	56.19	28.40	71.82	30.84	63.14	53.59
	V1+Filtering	**45.14**	**75.97**	78.62	44.78	51.50	55.23	25.80	**72.36**	31.21	64.48	**54.51**
	v1+init(wo imatrix)	41.55	73.57	79.94	44.87	50.75	56.30	**28.80**	71.11	32.19	64.01	54.30
	V1+Init	40.78	73.82	76.94	44.75	51.29	56.77	28.60	71.33	**32.07**	64.25	54.06
	Ours	43.52	75.13	74.89	**45.10**	**52.80**	**57.56**	26.00	71.22	31.70	**64.64**	54.26
Llama3.1-8B-I	V1	36.95	69.82	73.06	44.89	50.15	43.80	25.20	73.18	29.74	65.19	51.20
	V1+Filtering	36.86	69.99	70.89	46.72	55.48	46.05	**27.80**	72.31	**31.58**	65.75	52.34
	v1+init(wo imatrix)	37.97	70.45	78.32	46.89	57.73	47.07	27.00	72.52	29.01	66.46	53.34
	V1+Init	36.52	**71.76**	**80.21**	46.87	59.11	48.28	26.40	72.91	29.74	66.69	53.85
	Ours	**38.65**	71.21	77.46	**47.27**	**60.45**	**49.67**	26.20	**73.61**	29.87	**67.48**	**54.19**

Table 29: Ablation study of the calibration dataset for DeltaLoss at an average bit-width of 2.5 (options are W2A16G128 and W3A16G128), accounting for scale and zero-point bits. Tuning is employed.

Model	seqlen/samples	ARC-c	ARC-e	BoolQ	Hella	LAMB	MMLU	OBQA	PIQA	Truth.	Wino	GSM8K	AVG.
Qwen3-8B	256/16	42.06	74.20	79.30	45.09	53.93	61.85	27.60	71.98	30.97	64.88	51.86	54.88
	1024/128	43.43	75.88	82.48	45.47	54.43	63.12	27.60	72.63	30.84	68.19	59.21	56.94
Llama3.1-8B-I	256/16	42.66	74.20	81.56	49.61	63.42	57.11	27.40	75.57	30.11	70.24	31.69	54.87
	1024/128	42.92	74.58	83.12	49.61	66.58	57.42	27.20	75.73	28.52	69.93	31.92	55.23

H INT2/4 MIXED-BIT RESULTS

This appendix provides the detailed task-wise results for the INT2/4 mixed-bit weight-only quantization experiments discussed in Section 4.2. We report accuracies on ARC-Challenge, ARC-Easy, HellaSwag, PIQA, and WinoGrande for both uniform and mixed-precision settings across different model scales. These results complement the averaged comparisons in Table 1 and provide a more fine-grained view of the behavior of each method under extreme low-bit quantization. Specifically, Table 30 reports the detailed accuracies for Llama-3 models, and Table 31 reports the results for Llama-2 models.

Table 30: The detailed accuracies of 5 tasks for **Llama3 models** with INT2/4 mixed precision. "Avg. Bits" indicates average bit-width.

Model	Method	Avg. Bits	Group Size	ARC-c	ARC-e	Hella.	PIQA	Wino.	Avg.
Llama3-8B	16-bit	16	-	50.17	80.09	60.14	79.54	73.24	68.64
	AQLM	2	1x16	**41.21**	**74.24**	**55.44**	**77.80**	**71.82**	**64.10**
	EfficientQAT	2	128	36.01	69.15	50.74	75.30	65.67	59.37
	EfficientQAT	2	64	37.03	71.17	51.86	76.03	67.72	60.76
	SignRoundV1	2	128	34.22	67.09	44.15	71.65	59.12	55.25
	Ours	2	128	35.67	70.03	47.70	72.96	63.14	57.90
	Ours	2	64	37.37	70.29	48.69	74.59	66.14	59.42
	Ours*	2	128	38.99	72.18	49.19	73.94	65.57	59.97
	Ours*	2	64	39.42	71.42	50.04	74.43	66.61	60.38
	Ours	2.5	128	41.72	74.87	52.81	75.57	68.82	62.76
	Ours	2.5	64	**43.34**	**75.42**	53.27	76.06	**70.24**	**63.67**
	Ours*	2.5	128	41.38	74.87	53.31	75.46	67.96	62.60
	Ours*	2.5	64	43.00	74.87	**54.13**	**76.28**	69.53	63.56
Llama3-70B	16-bit	16	-	60.24	86.83	66.40	82.21	80.74	75.28
	AQLM	2	1x16	50.34	78.83	**63.47**	79.65	**78.22**	70.10
	EfficientQAT	2	128	48.81	79.25	60.75	79.60	69.46	67.57
	EfficientQAT	2	64	49.06	77.40	61.60	77.37	74.03	67.89
	SignRoundV1	2	128	47.87	79.71	43.96	78.62	73.64	64.76
	Ours	2	128	49.91	80.81	59.75	79.49	75.14	69.02
	Ours	2	64	52.22	81.73	60.93	**80.09**	75.69	70.13
	Ours*	2	128	52.05	82.41	60.41	79.38	76.56	70.16
	Ours*	2	64	**54.95**	**82.53**	61.05	79.82	76.32	**70.93**
	Ours	2.5	128	**56.83**	**83.67**	63.06	80.47	77.82	72.37
	Ours	2.5	64	55.29	83.08	62.60	81.23	77.66	71.97
	Ours*	2.5	128	56.48	83.04	**63.57**	**81.56**	**78.77**	**72.68**
	Ours*	2.5	64	56.40	82.95	63.07	80.63	78.53	72.32

Table 31: The detailed accuracies of 5 tasks for **Llama2 models** with INT2/4 mixed precision. "Avg. Bits" indicates average bit-width.

Model	Method	Avg. Bits	Group Size	ARC-c	ARC-e	Hella.	PIQA	Wino.	Avg.
Llama2-7B	16-bit	16	-	42.83	75.97	57.27	77.91	69.30	64.66
	AQLM	2	2x8	32.85	66.92	49.96	73.07	65.27	57.61
	AQLM	2	1x16	**39.68**	**74.07**	**53.42**	**76.88**	65.19	**61.85**
	QuIP#	2	-	37.88	71.84	52.19	75.46	65.67	60.61
	EfficientQAT	2	128	36.52	69.78	50.84	74.16	66.22	59.50
	EfficientQAT	2	64	36.86	70.96	51.58	75.30	65.98	60.14
	GPTQ	2	128	21.25	40.45	32.59	58.32	55.17	41.56
	AWQ	2	128	21.08	24.62	25.69	52.34	49.96	34.74
	OmniQ	2	128	23.46	50.13	40.28	65.13	55.88	46.98
	SignRoundV1	2	128	32.25	65.99	40.28	72.96	61.01	54.50
	Ours	2	128	33.53	69.32	48.17	74.43	63.93	57.88
	Ours	2	64	32.93	67.97	48.54	74.21	65.59	57.85
	Ours*	2	128	34.64	69.95	48.92	74.43	65.43	58.67
	Ours*	2	64	35.32	69.99	49.30	74.16	**66.41**	59.04
	Ours	2.5	128	35.41	71.46	50.62	75.30	67.01	59.96
	Ours	2.5	64	35.92	71.25	51.44	**75.35**	66.46	60.08
	Ours*	2.5	128	36.01	**72.01**	51.19	74.92	67.25	60.28
	Ours*	2.5	64	**36.09**	71.72	**51.85**	74.81	**67.72**	**60.44**
Llama2-13B	16-bit	16	-	47.18	78.32	60.25	79.22	72.22	67.44
	AQLM	2	2x8	40.10	73.06	54.62	77.09	66.22	62.22
	AQLM	2	1x16	**43.52**	75.25	**57.62**	**78.29**	**70.09**	**64.95**
	QuIP#	2	-	42.92	**75.72**	56.53	77.97	69.06	64.44
	EfficientQAT	2	128	42.83	75.04	55.66	76.99	68.90	63.88
	EfficientQAT	2	64	41.89	74.83	55.27	77.04	68.36	63.48
	GPTQ	2	128	21.93	55.60	41.06	67.08	55.80	48.29
	AWQ	2	128	23.12	26.22	25.80	52.99	51.85	35.99
	OmniQ	2	128	30.29	63.22	46.23	70.13	57.93	53.56
	SignRoundV1	2	128	38.57	71.17	53.35	76.17	64.33	60.72
	Ours	2	128	39.08	73.78	52.96	75.68	67.88	61.88
	Ours	2	64	40.53	74.33	53.87	76.44	68.43	62.72
	Ours*	2	128	40.44	74.24	53.15	75.90	67.96	62.34
	Ours*	2	64	40.78	74.58	53.73	76.77	68.19	62.81
	Ours	2.5	128	41.04	75.42	55.39	76.77	69.61	63.65
	Ours	2.5	64	41.13	75.55	55.81	**77.42**	70.56	64.09
	Ours*	2.5	128	40.02	75.08	55.61	76.82	**71.11**	63.73
	Ours*	2.5	64	**42.83**	**76.43**	**55.97**	77.37	70.24	**64.57**
Llama2-70B	16-bit	16	-	54.44	82.70	64.77	82.15	77.98	72.41
	AQLM	2	2x8	51.45	79.76	61.94	80.47	75.61	69.85
	AQLM	2	1x16	52.99	81.36	62.78	81.07	76.01	70.84
	QuIP#	2	-	52.65	**81.90**	**62.86**	**81.39**	75.77	**70.91**
	EfficientQAT	2	128	49.23	80.01	61.58	80.20	73.64	68.93
	EfficientQAT	2	64	50.77	80.13	61.78	80.14	74.59	69.48
	GPTQ	2	128	22.70	25.08	25.04	49.51	49.57	34.38
	AWQ	2	128	22.35	25.76	25.46	52.50	51.38	35.49
	OmniQ	2	128	33.28	67.21	35.45	74.10	64.33	54.87
	SignRoundV1	2	128	46.59	78.37	59.65	79.00	74.90	67.70
	Ours	2	128	49.06	79.59	59.99	78.35	74.98	68.39
	Ours	2	64	49.06	78.91	60.21	79.00	75.61	68.56
	Ours*	2	128	48.81	80.22	60.23	78.73	76.09	68.82
	Ours*	2	64	50.60	79.92	60.52	79.22	**76.24**	69.30
	Ours	2.5	128	**51.71**	80.56	61.77	79.82	76.64	70.10
	Ours	2.5	64	50.77	80.77	61.91	80.30	77.51	70.25
	Ours*	2.5	128	50.64	80.64	61.68	79.89	77.35	70.04
	Ours*	2.5	64	51.02	**81.31**	**62.29**	**80.58**	**77.82**	**70.60**