Understanding and Selecting Calibration Data for LLM Quantization: From Sensitivity Analysis to Activation-Based Curation

Maoxiong, Benben Hai, Anzheng Wang, Xiangdong Liu, Lingwei Meng
Tingting Wang, Shuli Zheng, Yixin Zhou, Haibiao Chen
AI Product Center
Kingsoft Corporation
Beijing, China
maoxiong@kingsoft.com

Abstract

Post-training quantization (PTQ) of large language models (LLMs) is highly sensitive to calibration data, yet what drives this sensitivity remains poorly understood. We present a systematic study spanning three PTQ algorithms—GPTQ, AWQ, and SmoothQuant+GPTQ—and six models from the Qwen2.5 and Llama-3.1 families, progressing from sensitivity analysis to activation-based data curation. Our sensitivity analysis jointly sweeps sample count and sequence length, revealing that their relative importance varies substantially across algorithms and model families, and identifying a robust combination that approaches saturated performance across both. We then show that domain-matched calibration data helps mainly for GPTQ on Qwen2.5 specialized variants, but provides little consistent benefit for AWQ, suggesting that AWQ's sensitivity is governed more by activation-distribution mismatch than by surface domain mismatch. Motivated by this diagnosis, we introduce the Activation Centroid Distance Metric (ACDM), a curation method that selects calibration samples by aligning their per-layer activation statistics with task-reference centroids estimated from held-out training-side samples, directly targeting the quantity that domain labels fail to capture. ACDM improves both average accuracy and cross-task balance over three baselines—random sampling, self-calibration, and ZipCal—across five benchmarks, achieving gains of $+0.95$ and $+0.49$ percentage points over random sampling on Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct, respectively. We release an automated toolkit at https://anonymous.4open.science/r/adaptive-quant-toolkit-D43E.

1 Introduction

Large language models (LLMs) have demonstrated impressive performance across a wide variety of tasks (Dubey et al., 2024; Hui et al., 2024), but their deployment remains expensive (Wan et al., 2023). Post-training quantization (PTQ) (Frantar et al., 2022; Lin et al., 2024; Xiao et al., 2023; Ashkboos et al., 2024) provides a method to compress models without the need for retraining. However, its accuracy heavily relies on a small set of unlabeled *calibration* samples used to estimate activation statistics (Williams and Aletras, 2024; Gong et al., 2024; He et al., 2025). Despite the critical role of calibration data, several aspects of calibration sensitivity remain insufficiently explored.

Prior research typically examines the impact of sample size and sequence length in isolation (Williams and Aletras, 2024; Dettmers et al., 2023), overlooking their *joint* influence across different PTQ algorithms and model architectures. Moreover, while domain-specialized models exhibit significant differences in weight distribution compared to general-purpose models (Hui et al., 2024; Yang et al., 2024; Ping et al., 2024), the effect of domain-matched calibration data for specialized models across various architectures has not been thoroughly investigated. Additionally, existing calibration data selection methods exhibit fundamental limitations in addressing PTQ-specific needs. Self-calibration (Williams et al., 2025) generates synthetic data to approximate the pre-training distribution, but focuses exclusively on the model's output-side generation behavior while neglecting the alignment of internal activation distributions with evaluation-side inputs—a property particularly critical for AWQ. ZipCal (Monaco et al., 2026) selects data based on token frequency statistics following a Zipfian distribution, but is entirely model-agnostic and blind to how different calibration samples drive layer-wise activations, making it unable to account for the distinct optimization mechanisms of GPTQ and AWQ.

We address these gaps through a comprehensive study that investigates the interaction between model families (Qwen2.5 (Qwen Team, 2025),

Llama-3.1 (Dubey et al., 2024)), specialization levels (general, code, math), benchmarks, and PTQ algorithms. Our contributions are as follows:

- We propose a unified framework connecting dataset analysis to data selection, offering a systematic approach to understanding calibration data sensitivity across both data characteristics and selection strategies.
- We conduct a large-scale sensitivity analysis over 1,000+ quantization-and-evaluation runs across 6 models, 3 PTQ algorithms, and 5 benchmarks, investigating the *joint* effects of sample size and sequence length, and systematically examining the role of domain-matched calibration across model families, specialization levels, and PTQ algorithms.
- We introduce the Activation Centroid Distance Metric (ACDM), a parameter-training-free and label-free data selection method that ranks calibration samples by aligning their per-layer activation statistics with task-reference centroids estimated from training-side held-out samples. By operating directly in activation space, ACDM targets AWQ's activation-distribution sensitivity and improves both quantization performance and cross-task balance.

The remainder of this paper is organized as follows. Section 2 reviews related work on LLM quantization and calibration-data selection. Section 3 presents the calibration sensitivity analysis. Section 4 introduces ACDM and reports experimental results. Section 5 concludes the paper. The Limitations section discusses limitations, and Appendix A provides full quantization results.

2 Related Work

2.1 LLM Quantization

Post-training quantization (PTQ) reduces model precision while preserving performance. GPTQ (Frantar et al., 2022) minimizes reconstruction error with Hessian-based quantization; AWQ (Lin et al., 2024) rescales weights using activation magnitudes; and SmoothQuant (Xiao et al., 2023) shifts quantization difficulty from activations to weights. Mixed-precision methods such as SpQR (Dettmers et al., 2023) further improve efficiency. Because these methods use calibration data differently, they also differ in calibration sensitivity, which motivates our analysis.

2.2 Calibration Data Selection

Existing calibration-data selection methods mainly fall into two categories. Generation-based methods, such as self-calibration (Williams et al., 2025), synthesize calibration data from model outputs. Compression-based methods, such as ZipCal (Monaco et al., 2026), select samples using lexical diversity or related surface-form criteria. However, neither explicitly targets the activation distributions that drive quantization. In contrast, our ACDM selects samples by directly measuring per-layer activation similarity to the target evaluation distribution.

3 Calibration Sensitivity Analysis

We diagnose calibration sensitivity along two orthogonal axes: *format* (sample count, sequence length) and *content* (which samples). The first axis localizes a robust hyperparameter operating point; the second exposes the residual sensitivity that motivates the activation-space selection method in Section 4.

3.1 Setup

We pair two model families to disentangle architecture from domain specialization: Qwen2.5-7B-Instruct, Qwen2.5-Coder-7B-Instruct, and Qwen2.5-Math-7B-Instruct from the Qwen2.5-7B family (Qwen Team, 2025); and Llama-3.1-8B-Instruct, Llama-3.1-Pro-Coder,[1] and Llama-3.1-Mather[2] from the Llama-3.1 8B family (Dubey et al., 2024), covering general, code, and math specializations in both families. We evaluate on GSM8K (Cobbe et al., 2021), HumanEval (Chen, 2021), MBPP (Austin et al., 2021), MMLU (Hendrycks et al., 2020), and CMMLU (Li et al., 2024), averaging three runs per configuration. Calibration data is drawn from General (UltraChat (Ding et al., 2023)), Code (CodeSearchNet (Husain et al., 2019)), and Math (MathQA-Python (Amini et al., 2019)) sources; the domain-alignment analysis (Section 3.3) uses unlabeled task-specific test inputs solely to diagnose domain correspondence, not as a deployment protocol.

[1] https://huggingface.co/HemanthKari/Llama-3.1-Pro-Coder-v1

[2] https://huggingface.co/Neelectric/Llama-3.1-8B-Instruct_SFT_mathfisher_v00.03

Quantization is performed with `llmcompressor`[3] and evaluation with `lm-evaluation-harness`.[4] AWQ runs at w4a16 for both families; GPTQ at w4a16 for Qwen2.5 and w8a16 for Llama-3.1 (w4a16 causes catastrophic collapse on coding benchmarks, Table 12); and SQ+GPTQ at w8a8. This precision asymmetry makes cross-family comparisons indicative rather than strictly controlled. Full results appear in Appendices A.1–A.2 and Figure 7.

3.2 Hyperparameter Sensitivity

We sweep the sample count n over the candidate set $\{4, 8, 16, 32, 64, 128, 256\}$ and sequence length ℓ over $\{64, 512, 1024, 2048\}$, decomposing score variance via η^2 analysis. Figures 1–2 show the full (n, ℓ) score grids; Figures 3–4 report the η^2 decomposition.

Algorithm sensitivity is family-dependent. On Qwen2.5-7B-Instruct, GPTQ is highly volatile: its mean5 score (the unweighted average over five benchmarks) spans 0.1025 across calibration settings, corresponding to up to 18% degradation below FP16 under poor choices; AWQ and SQ+GPTQ are far more stable, with score ranges of approximately 0.04 (Figure 7). On Llama-3.1-8B-Instruct—*under the operational precisions used here* (AWQ w4a16, GPTQ w8a16, and SQ+GPTQ w8a8)—all three algorithms vary by less than 0.013, suggesting that format-induced volatility is substantially suppressed under these precision settings. Note that GPTQ at w4a16 is substantially *more* volatile on Llama-3.1-8B-Instruct than on Qwen2.5-7B-Instruct (Table 12); the observed robustness is therefore precision-conditional, not purely architectural.

Sequence length dominates on Qwen2.5-7B-Instruct, but not on Llama-3.1-8B-Instruct. On Qwen2.5-7B-Instruct, the η^2 decomposition attributes 75.2% of AWQ's variance to sequence length, with sample count below 4%. AWQ saturates quickly—even n=4 at ℓ=512 nearly matches the global optimum—while short sequences cause consistent degradation. GPTQ shows a more balanced sensitivity, with sequence length accounting for 42.2% of the variance and sample count accounting for 15.8%, making it the only algorithm where joint tuning of (n, ℓ) meaningfully matters. On Llama-3.1-8B-Instruct, residual variance dominates all three algorithms (57.5–67.7%) and sequence length has near-negligible explanatory power, suggesting that factors beyond sample count and sequence length, including sample identity and other unmodeled effects, play a larger role. This observation directly motivates Section 3.3.

Operating point. (n=256, ℓ=1024) emerges as a robust cross-family default under AWQ: it is within 0.012 of the empirical optimum on Qwen2.5-7B-Instruct and within 0.004 on Llama-3.1-8B-Instruct. We fix this setting throughout Section 4 so that observed differences reflect sample *content* rather than format.

Table 1: Qwen2.5-7B variants with in-distribution calibration. Bold: best per metric within each algorithm group.

Algo.	Calib.	GSM8K	HumanEval	MMLU
Qwen2.5-7B-Instruct				
GPTQ	gsm8k	.7715	.5793	.7126
GPTQ	humaneval	**.7775**	**.6402**	.7090
GPTQ	mmlu	.7460	.5732	**.7162**
AWQ	gsm8k	.7830	.5854	**.7105**
AWQ	humaneval	**.7833**	.5869	.7083
AWQ	mmlu	.7673	**.6199**	.7090
SQ+GPTQ	gsm8k	.8021	**.6341**	**.7159**
SQ+GPTQ	humaneval	**.8029**	.6220	.7151
SQ+GPTQ	mmlu	.7953	.6159	.7154
Qwen2.5-Coder-7B-Instruct				
GPTQ	gsm8k	**.7755**	.7012	.6388
GPTQ	humaneval	.7460	**.7988**	**.6424**
GPTQ	mmlu	.7660	.7622	.6409
AWQ	gsm8k	.7490	**.7805**	.6444
AWQ	humaneval	**.7690**	**.7805**	**.6455**
AWQ	mmlu	.7570	.7439	**.6455**
SQ+GPTQ	gsm8k	.7710	**.8293**	**.6438**
SQ+GPTQ	humaneval	.7824	.8171	.6422
SQ+GPTQ	mmlu	**.7900**	.8171	.6431
Qwen2.5-Math-7B-Instruct				
GPTQ	gsm8k	**.8810**	.4695	.5311
GPTQ	humaneval	.8770	**.5061**	.5309
GPTQ	mmlu	.8805	.4695	**.5360**
AWQ	gsm8k	.8850	.4756	**.5365**
AWQ	humaneval	**.8910**	**.4878**	.5358
AWQ	mmlu	.8830	.4634	.5345
SQ+GPTQ	gsm8k	.8923	.4817	.5375
SQ+GPTQ	humaneval	**.8969**	.4695	**.5396**
SQ+GPTQ	mmlu	.8817	**.4878**	.5381

[3] https://github.com/vllm-project/llm-compressor

[4] https://github.com/EleutherAI/lm-evaluation-harness

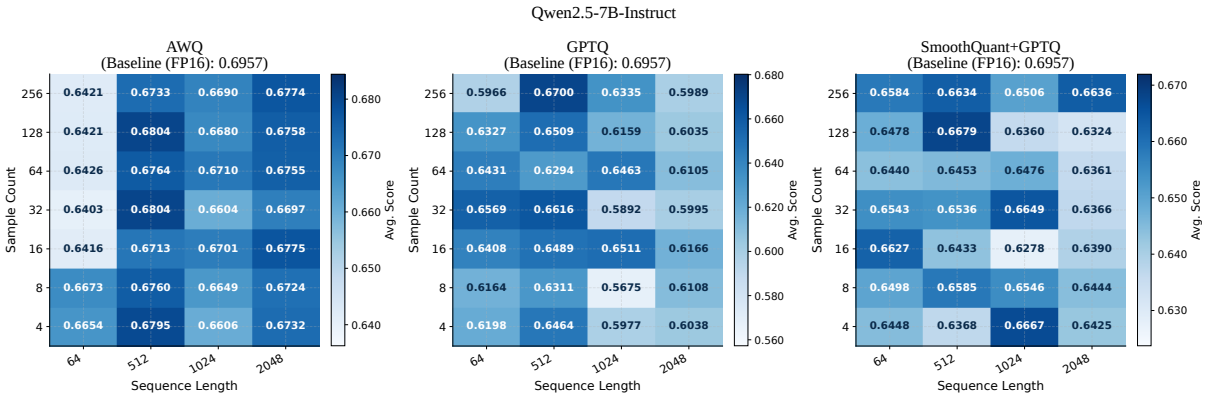

Figure 1: Qwen2.5-7B-Instruct: mean score across sample count and sequence length under three PTQ algorithms. Each cell averages five benchmarks and three runs.

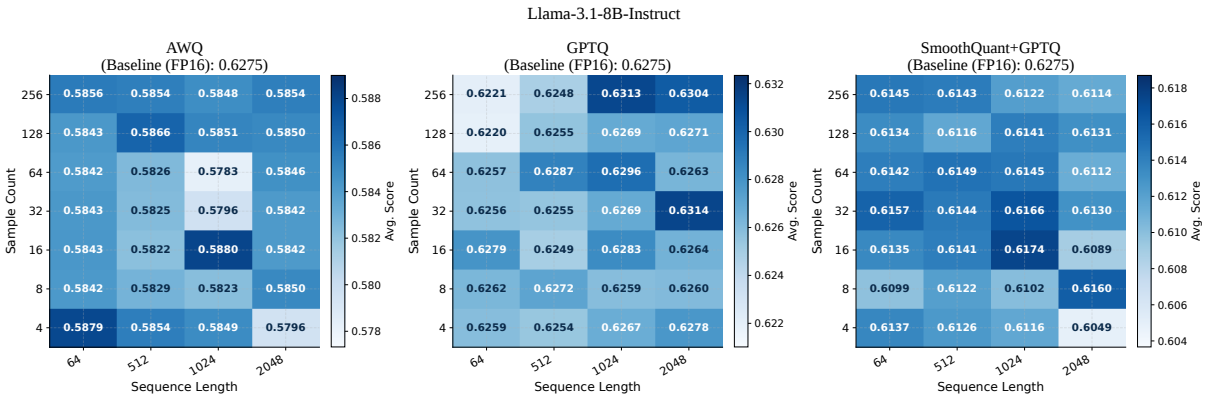

Figure 2: Llama-3.1-8B-Instruct: same setup as Figure 1. Score ranges are an order of magnitude tighter; GPTQ uses w8a16 because w4a16 collapses coding benchmarks on Llama-3.1-8B-Instruct, as shown in Table 12.

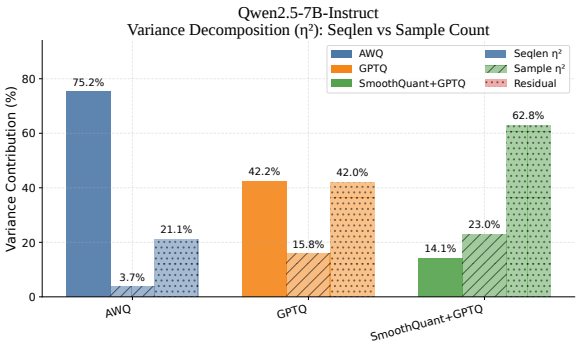

Figure 3: η^2 decomposition for Qwen2.5-7B-Instruct. Sequence length dominates AWQ (75.2%).

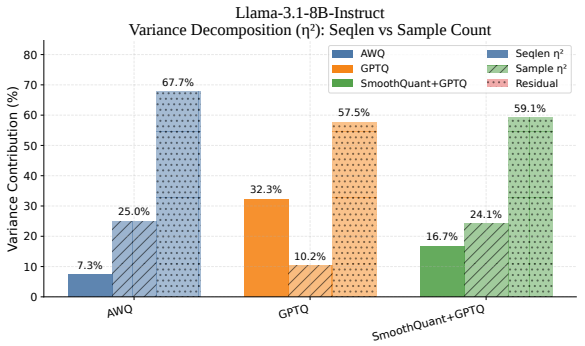

Figure 4: η^2 decomposition for Llama-3.1-8B-Instruct. Residuals dominate all three algorithms ($> 57\%$).

3.3 Domain Alignment and Its Limits

If hyperparameters do not fully explain the residual variance observed on Llama-3.1-8B-Instruct, the remaining sensitivity is likely driven by sample content and other unmodeled factors. We test the most natural content-side hypothesis—that domain-matched calibration benefits specialized models—by defining *domain alignment*: a specialized model attains its best native-benchmark score when calibrated with matching-domain data.

Domain alignment is narrow. The pattern holds clearly only for GPTQ on the Qwen2.5 specialized models (Table 1): Qwen2.5-Coder-7B-Instruct reaches its best HumanEval score (0.7988) under HumanEval calibration, and Qwen2.5-Math-7B-Instruct reaches its best GSM8K score (0.8810)

Table 2: Llama-3.1-8B variants with in-distribution calibration. SQ+GPTQ: w8a8; GPTQ: w8a16; AWQ: w4a16. Bold: best per metric within each algorithm group.

Algo.	Calib.	GSM8K	HumanEval	MMLU
Llama-3.1-8B-Instruct				
AWQ	gsm8k	.7089	**.5915**	.6227
AWQ	humaneval	.7263	.5366	.6224
AWQ	mmlu	**.7293**	.5793	**.6259**
GPTQ	gsm8k	.7915	**.6037**	**.6318**
GPTQ	humaneval	.7908	.5915	.6317
GPTQ	mmlu	**.7953**	**.6037**	.6312
SQ+GPTQ	gsm8k	**.7733**	**.5793**	.6356
SQ+GPTQ	humaneval	**.7733**	.5732	.6293
SQ+GPTQ	mmlu	.7612	.5610	**.6365**
Llama-3.1-Pro-Coder				
AWQ	gsm8k	.6588	**.5244**	.5956
AWQ	humaneval	.6626	**.5244**	**.5983**
AWQ	mmlu	**.6694**	.4939	.5981
GPTQ	gsm8k	.6823	**.4756**	**.6243**
GPTQ	humaneval	**.6945**	**.4756**	.6240
GPTQ	mmlu	.6808	.4634	.6231
SQ+GPTQ	gsm8k	.6808	.4329	.6228
SQ+GPTQ	humaneval	**.6854**	**.5000**	**.6239**
SQ+GPTQ	mmlu	.6710	.4817	.6212
Llama-3.1-Mather				
AWQ	gsm8k	.7976	**.1646**	.2688
AWQ	humaneval	.7938	.1585	**.2720**
AWQ	mmlu	**.7998**	.1524	.2697
GPTQ	gsm8k	.8082	.1951	.2627
GPTQ	humaneval	.8089	**.2134**	.2634
GPTQ	mmlu	**.8135**	.2012	**.2639**
SQ+GPTQ	gsm8k	.8105	**.2134**	.2589
SQ+GPTQ	humaneval	**.8203**	.1951	.2607
SQ+GPTQ	mmlu	.8158	**.2134**	**.2614**

under GSM8K calibration. On the Llama-3.1 specialized models, however, the pattern is inconsistent (Table 2). For Llama-3.1-Pro-Coder, the best HumanEval score ties across calibration sets under AWQ and GPTQ, although SQ+GPTQ favors HumanEval calibration. For Llama-3.1-Mather, the best GSM8K score consistently comes from non-matching calibration, with the exact source depending on the algorithm. This suggests that Llama-3.1 fine-tuning yields weaker domain-separable structure in quantization-relevant statistics.

AWQ is domain-insensitive in both families. Across all six models, AWQ shows only a narrow score range over calibration domains—for example, GSM8K spans 0.7673–0.7833 on Qwen2.5-7B-Instruct and 0.7089–0.7293 on Llama-3.1-8B-Instruct—and the best native-task scores frequently come from non-matching calibration data. SQ+GPTQ behaves similarly. Domain matching therefore does not explain the residual variance observed in Section 3.2, nor does it account for a meaningful share of AWQ's behavior in either family.

Toward activation-distributional selection. These results clarify where the problem remains open. For specialized Qwen2.5 models under GPTQ, domain-matched calibration already provides a meaningful signal, leaving limited room for further improvement. The harder case is the general-purpose instruct model under AWQ: domain labels provide weak and inconsistent guidance regardless of model family, yet calibration choice still affects performance. This is consistent with AWQ's sensitivity being driven more by activation-distribution mismatch than by surface domain mismatch (Lin et al., 2024). We therefore focus Section 4 on this setting.

4 ACDM-Based Calibration Data Selection

Building on the sensitivity analysis in Section 3, which shows that calibration sensitivity is not fully explained by hyperparameters or domain labels, we propose ACDM (Activation Centroid Distance Metric), a lightweight method that ranks calibration samples by the alignment of their per-layer activation statistics with task-reference activation centroids. Unlike domain-label-based curation, ACDM operates directly in activation space and is designed to preserve performance across heterogeneous tasks rather than over-specializing calibration to a single benchmark. As illustrated in Figure 5, ACDM compares per-layer activation statistics of candidate samples against task-reference centroids estimated from held-out training-side samples and retains the closest ones.

4.1 Activation-Centroid Distance Metric

For layer l, sample i is summarized by its activation mean $\mu_i^{(l)} = \mathbb{E}[A_i^{(l)}]$ and standard deviation $\sigma_i^{(l)} = \mathrm{Std}[A_i^{(l)}]$. For task k, the task-reference centroid $\mathcal{C}_k^{(l)} = (\bar{\mu}_k^{(l)}, \bar{\sigma}_k^{(l)})$ is estimated from N held-out training-side samples disjoint from the final evaluation set:

$$\bar{\mu}_k^{(l)} = \frac{1}{N}\sum_{i=1}^{N} \mu_i^{(l)}, \quad \bar{\sigma}_k^{(l)} = \frac{1}{N}\sum_{i=1}^{N} \sigma_i^{(l)}. \quad (1)$$

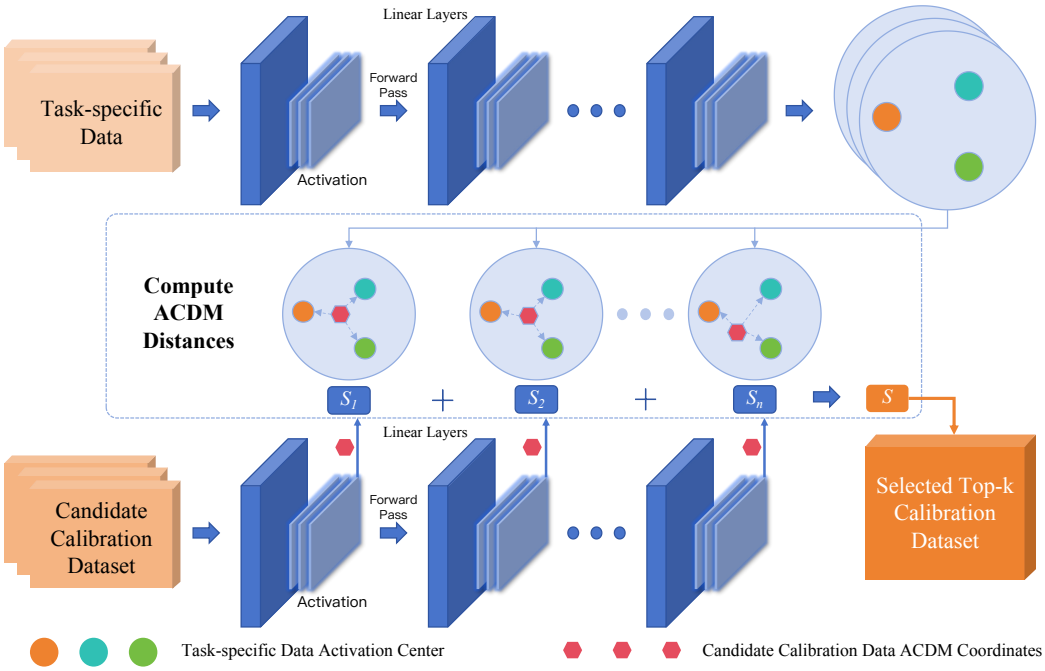

Figure 5: Overview of the ACDM pipeline. Task-related reference samples from representative training-side splits are passed through the model to compute per-layer activation statistics and estimate task-reference centroids (top branch). Candidate calibration samples undergo the same forward pass (bottom branch) and are scored by their average distance to these centroids across all layers and tasks. The closest-matching samples are retained as the final calibration set.

The distance from candidate sample s to centroid $\mathcal{C}_k^{(l)}$ at layer l is

$$D^{(l)}(\mathcal{C}_k, s) = \|\bar{\mu}_k^{(l)} - \mu_s^{(l)}\|_2^2 + \gamma \, \|\bar{\sigma}_k^{(l)} - \sigma_s^{(l)}\|_2^2, \tag{2}$$

with $\gamma \geq 0$ controlling the relative weight of standard-deviation alignment. The aggregate score averages over L layers and K tasks:

$$\mathrm{ACDM}(s) = \frac{1}{KL} \sum_{k=1}^{K} \sum_{l=1}^{L} D^{(l)}(\mathcal{C}_k, s). \tag{3}$$

The top-B samples with the lowest ACDM scores form the calibration set, where $B = 256$ in our experiments.

4.2 Quantization-Aware Rationale for ACDM

AWQ (Lin et al., 2024) determines per-channel scaling factors from calibration activations, with $\mathbf{r}_c \propto \mathbb{E}[|x_c|]^{\alpha}$, to reduce the layer-wise weight-quantization error $\|\mathbf{Q}(\mathbf{W}\,\mathrm{diag}(\mathbf{r})) - \mathbf{W}\,\mathrm{diag}(\mathbf{r})\|_F$. When the calibration activation distribution deviates from the target-task distribution, the estimated scale $\hat{\mathbf{r}}$ may become suboptimal at inference time.

Approximating the scale mismatch by first-order deviations in activation statistics, we formulate the following proxy objective for the total quantization error across tasks and layers:

$$\epsilon_{\text{total}} \lesssim \sum_{k=1}^{K} \sum_{l=1}^{L} \alpha_l \Big(\|\bar{\mu}_k^{(l)} - \mu_s^{(l)}\|_2^2 + \gamma \, \|\bar{\sigma}_k^{(l)} - \sigma_s^{(l)}\|_2^2 \Big), \tag{4}$$

where $\alpha_l > 0$ is a layer-wise sensitivity weight and γ balances the mean and spread terms. Under uniform layer weights ($\alpha_l = L^{-1}$), Equation (3) minimizes this proxy objective. Thus, ACDM selects calibration samples whose activation statistics better match the target task suite.

4.3 Evaluation Protocol for ACDM

We evaluate Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct under AWQ w4a16 with the 256×1024 configuration, reporting unweighted mean5. Task-reference centroids use N=128 held-out samples from GSM8K train (math), MBPP train (code), and MMLU auxiliary train (general knowledge), covering all three evaluation domains without assuming universal training splits. S-G draws 30k general-domain samples from UltraChat; S-mixed draws equally from UltraChat, CodeSearchNet, and MathQA-Python. Baselines are Random,

Table 3: Mean5 of calibration-data selection methods under two candidate pools (AWQ w4a16, 256×1024). The full fixed γ grid is reported to expose sensitivity rather than hiding hyper-parameter selection. **Bold**: global best per model; underline: pool-level best when it differs from the global best.

Type	Strategy	Qwen2.5-7B Instruct	Llama-3.1-8B Instruct
S-G (general-domain, 30k)			
Random	Random	.6690±.0021	.5848±.0008
Generation	Self-cal	.6442±.0013	.5857±.0007
Compression	ZipCal	.6689±.0011	.5871±.0008
ACDM	γ=0.6	.6756±.0003	.5876±.0015
	γ=0.8	.6739±.0007	.5885±.0018
	γ=1.0	.6665±.0003	**.5897**±.0010
	γ=1.2	.6685±.0016	.5888±.0046
	γ=1.4	.6706±.0007	.5890±.0013
S-mixed (cross-domain, 30k)			
Random	Random	.6693±.0016	.5855±.0011
Generation	Self-cal	.6442±.0013	.5857±.0007
Compression	ZipCal	.6709±.0013	.5866±.0009
ACDM	γ=0.6	.6721±.0010	.5889±.0018
	γ=0.8	**.6788**±.0005	.5878±.0012
	γ=1.0	.6718±.0005	.5852±.0001
	γ=1.2	.6716±.0022	.5887±.0023
	γ=1.4	.6725±.0010	.5879±.0003

Self-calibration (Williams et al., 2025), and ZipCal (Monaco et al., 2026); for each γ, ACDM scores the pool via (3) and retains the 256 lowest-scoring samples. Each cell reports mean±std over three calibration seeds.

4.4 Main Results of ACDM

Table 3 reports mean5 under both candidate pools; Self-calibration results are identical across blocks because the calibration data is generated from the model itself and is therefore independent of the candidate pool.

Aggregate performance. ACDM delivers the strongest aggregate performance under the fixed grid. On Qwen2.5-7B-Instruct, it reaches .6756 (γ=0.6) on S-G and the global best .6788 (γ=0.8) on S-mixed, a +0.95 pp gain over Random. On Llama-3.1-8B-Instruct, ACDM attains the global best .5897 on S-G (γ=1.0), a +0.49 pp gain over Random, and the pool-level best .5889 on S-mixed (γ=0.6). Averaged over the five γ values, ACDM exceeds the strongest non-ACDM baseline in all four model–pool blocks and wins in 17 of 20 model–pool–γ settings.

Cross-task balance. Figure 6 provides a visual summary of the task-level performance profiles, while Table 4 reports the corresponding per-benchmark scores. Across the five individual benchmark metrics, ACDM is best or tied-best on three metrics for each model block. The baselines show more localized gains: ZipCal is strong on selected axes such as GSM8K for Llama-3.1-8B-Instruct and HumanEval for Qwen2.5-7B-Instruct, while Self-calibration attains the best GSM8K score on Qwen2.5-7B-Instruct but drops sharply on MBPP. These results suggest that ACDM maintains a more balanced task-level profile, rather than relying on a single-task gain.

Table 4: Single-column companion table for Figure 6. Each block reports the per-benchmark scores of each method's best overall configuration under AWQ w4a16 and 256×1024. **Bold** marks the best score within each model block and metric column; ties are both bolded.

Method	CMMLU	MMLU	MBPP	HE	GSM8K
Llama-3.1-8B-Instruct					
ACDM	.5147	**.6342**	**.5220**	**.5671**	.7104
ZipCal	**.5152**	.6330	.5160	.5488	**.7225**
Self-cal	.5130	.6312	**.5220**	.5549	.7074
Random	.5138	.6306	.5193	.5589	.7048
Qwen2.5-7B-Instruct					
ACDM	**.7803**	**.7127**	**.5400**	.5854	.7756
ZipCal	.7753	.7038	.5180	**.5915**	.7665
Self-cal	.7752	.7010	.3820	.5854	**.7779**
Random	.7773	.7037	.5167	.5833	.7652

Candidate-pool effects. The optimal candidate pool is model-dependent: Qwen2.5-7B-Instruct benefits more from the cross-domain pool, whereas Llama-3.1-8B-Instruct prefers the homogeneous general-domain pool, suggesting that domain diversity alone is not sufficient; activation-level coherence with the target model is more predictive than explicit domain composition.

4.5 Sensitivity to γ

The best γ varies across models and pools: Qwen2.5-7B-Instruct prefers γ=0.6 on S-G and γ=0.8 on S-mixed, while Llama-3.1-8B-Instruct prefers γ=1.0 on S-G and γ=0.6 on S-mixed. However, γ is not the main driver of the result. Only 3 of 20 ACDM settings fall below the strongest non-ACDM baseline, and the γ-averaged score remains above that baseline in every model–pool block. Thus, γ affects the exact optimum, but the central conclusion—activation-based selection improves both mean performance and cross-task balance—does not rely on a single favorable value.

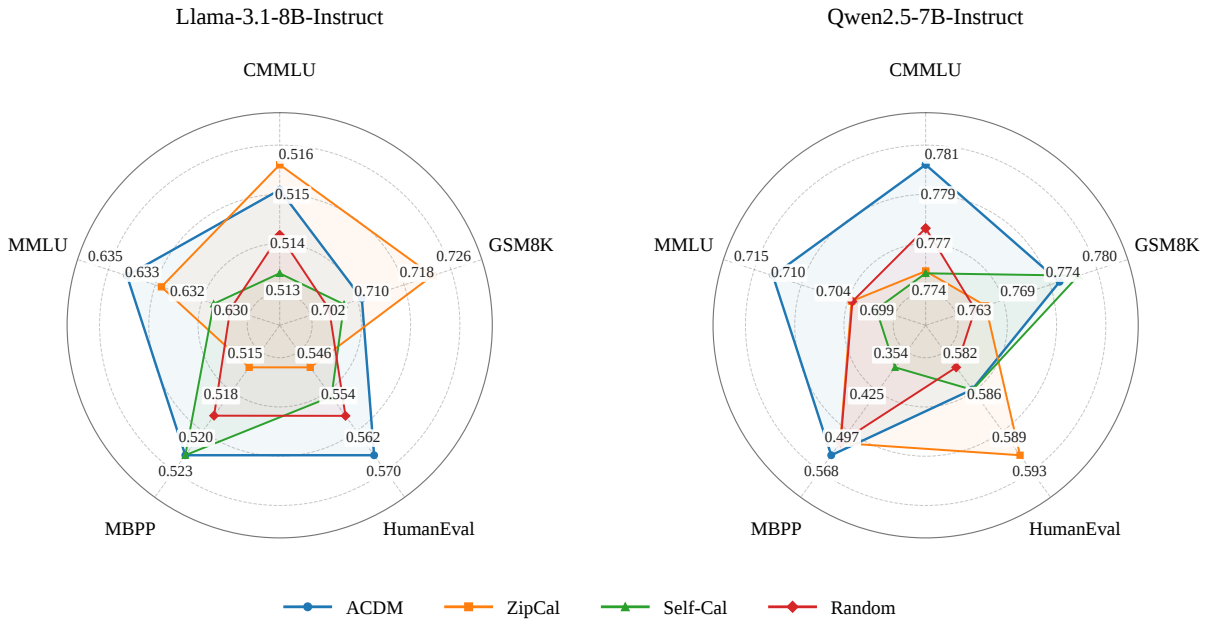

Figure 6: Per-benchmark performance of each method's best overall configuration across candidate pools and, for ACDM, across the fixed γ grid. Results are shown for Llama-3.1-8B-Instruct (left) and Qwen2.5-7B-Instruct (right). ACDM yields a balanced profile across all five tasks, whereas baselines exhibit more pronounced task-level asymmetries.

4.6 Key Findings for ACDM

ACDM achieves the global best mean5 in both model columns: Qwen2.5-7B-Instruct on S-mixed (γ=0.8, .6788±.0005) and Llama-3.1-8B-Instruct on S-G (γ=1.0, .5897±.0010). Its grid-level behavior is also stronger than the baselines: averaged over γ, ACDM exceeds the strongest non-ACDM method in all four model–pool blocks and wins in 17 of 20 individual settings. More importantly, the radar chart and the single-column per-benchmark table show that ACDM preserves a more balanced task profile after quantization, avoiding the single-task gains and cross-task collapses observed in ZipCal and Self-calibration. Together, these results indicate that effective calibration selection depends on activation-level representativeness for the target model, rather than explicit domain labels or post-hoc hyper-parameter choice.

5 Conclusion

This work studies how calibration data affects post-training quantization (PTQ) for large language models, progressing from sensitivity analysis to a calibration selection framework for balanced quantized performance. We show that calibration sensitivity is highly model- and algorithm-dependent: sequence length dominates AWQ on Qwen2.5-7B-Instruct (75.2% of variance), while residual factors prevail on Llama-3.1-8B-Instruct (67.7%). Domain matching explains only part of this behavior, helping mainly for GPTQ on specialized Qwen2.5 variants but providing little consistent benefit for AWQ. This suggests that AWQ is governed less by surface domain labels than by activation-distribution mismatch. Motivated by this diagnosis, we introduce ACDM, a parameter-training-free and label-free calibration selection method that aligns candidate samples with task-related activation centroids. By operating on per-layer activation statistics, ACDM directly targets the distributional signal that drives AWQ's weight scaling. Empirically, ACDM outperforms random sampling, Self-calibration, and ZipCal, improving mean5 by $+0.95$ pp on Qwen2.5-7B-Instruct and $+0.49$ pp on Llama-3.1-8B-Instruct over random sampling. More importantly, it preserves a more balanced cross-task performance profile, avoiding the per-benchmark trade-offs observed in the baselines. Together, these findings provide evidence that activation-aware calibration selection is a promising direction for improving both average accuracy and performance balance in LLM PTQ.

Limitations

Our study has several limitations that warrant consideration for future work.

Model Scale. Our experiments focus on 7B/8B-parameter models from two families (Qwen2.5 and Llama-3.1). While this design enables controlled comparison across architectures and domain specializations, it does not examine how model scale affects calibration sensitivity. Larger models (e.g., 14B, 72B parameters) may exhibit different activation distributions and quantization behaviors, and the pronounced architecture-dependent patterns we observe (e.g., Qwen2.5's high sensitivity vs. Llama-3.1's robustness) may not hold at different scales.

Computational overhead of ACDM. ACDM requires a full forward pass over all candidate samples to extract per-layer activation statistics, incurring approximately 30 minutes on a 30k-sample pool with a 7B-parameter model on two NVIDIA H20 GPUs. This is substantially slower than random sampling or lexical methods such as ZipCal. ACDM is therefore best suited to settings where calibration quality is prioritized over selection speed, or where the selection cost can be amortized across multiple quantization runs.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. pages 2357–2367.

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

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, and Quoc Le. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Mark Chen. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

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

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

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. pages 3029–3051.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, and Angela Fan. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

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

Ruihao Gong, Yang Yong, Shiqiao Gu, Yushi Huang, Yunchen Zhang, Xianglong Liu, and Dacheng Tao. 2024. Llm-qbench: A benchmark towards the best practice for post-training quantization of large language models.

Bowei He, Lihao Yin, Huiling Zhen, Shuqi Liu, Han Wu, Xiaokun Zhang, Mingxuan Yuan, and Chen Ma. 2025. Preserving llm capabilities through calibration data curation: From analysis to optimization. *arXiv preprint arXiv:2510.10618*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, and Keming Lu. 2024. Qwen2.5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. 2019. Codesearchnet challenge: Evaluating the state of semantic code search. *arXiv preprint arXiv:1909.09436*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024. Cmmlu: Measuring massive multitask language understanding in chinese. pages 11260–11285.

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

Francesco Pio Monaco, Elia Cunegatti, Flavio Vella, and Giovanni Iacca. 2026. Frequency matters: Fast model-agnostic data curation for pruning and quantization. *arXiv preprint arXiv:2603.16105*.

Bowen Ping, Shuo Wang, Hanqing Wang, Xu Han, Yuzhuang Xu, Yukun Yan, Yun Chen, Baobao Chang,

Zhiyuan Liu, and Maosong Sun. 2024. Delta-come: Training-free delta-compression with mixed-precision for large language models. volume 37, pages 31056–31077.

Qwen Team. 2025. Qwen2.5 technical report.

Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, and Quanlu Zhang. 2023. Efficient large language models: A survey. *arXiv preprint arXiv:2312.03863*.

Miles Williams and Nikolaos Aletras. 2024. On the impact of calibration data in post-training quantization and pruning. pages 10100–10118.

Miles Williams, George Chrysostomou, and Nikolaos Aletras. 2025. Self-calibration for language model quantization and pruning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10149–10167.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. pages 38087–38099.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, and Junyang Lin. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

A Appendix

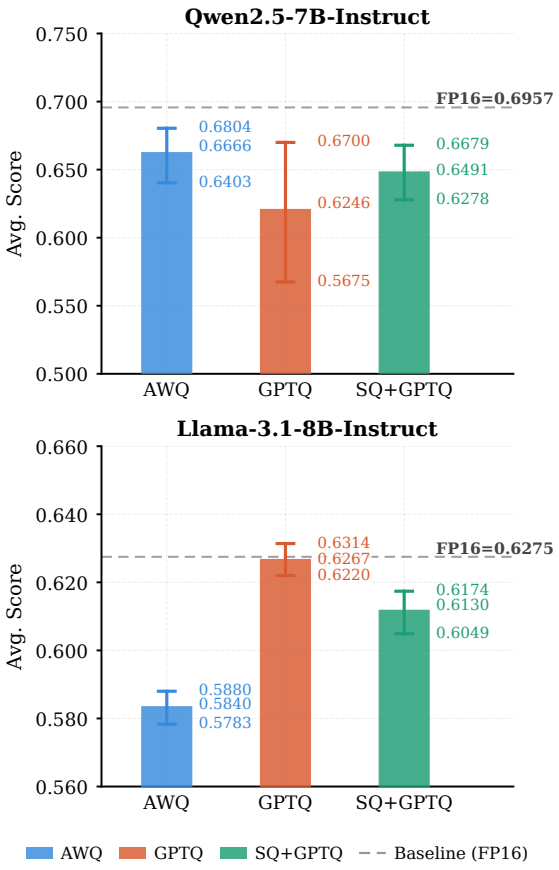

Figure 7: Algorithm-level summary. Bars show the mean score (five benchmarks × all calibration configurations); whiskers mark the min/max configuration-level scores, not error bars.

A.1 Quantization Results for Qwen2.5-7B-Instruct

Table 5: Qwen2.5-7B-Instruct with G calibration: AWQ (w4a16 Quantization).

n	*ell*	CMM	MML	MBP	HE	GSM
4	64	.776	.710	.457	.596	.789
4	512	.777	.714	.520	.616	.771
4	1024	.779	.711	.485	.575	.752
4	2048	.778	.715	.500	.608	.766
8	64	.775	.710	.484	.592	.776
8	512	.777	.712	.528	.598	.765
8	1024	.777	.714	.491	.594	.749
8	2048	.776	.714	.527	.585	.761
16	64	.778	.710	.418	.552	.751
16	512	.777	.714	.516	.598	.752
16	1024	.777	.713	.513	.598	.750
16	2048	.776	.713	.539	.596	.764
32	64	.778	.710	.417	.546	.751
32	512	.776	.713	.550	.619	.744
32	1024	.777	.713	.486	.573	.752
32	2048	.778	.713	.502	.595	.762
64	64	.778	.710	.426	.549	.751
64	512	.777	.712	.520	.604	.769
64	1024	.777	.713	.514	.598	.754
64	2048	.776	.713	.535	.585	.767
128	64	.777	.710	.416	.558	.750
128	512	.776	.713	.550	.619	.745
128	1024	.776	.713	.505	.602	.744
128	2048	.776	.713	.539	.585	.766
256	64	.778	.710	.425	.547	.751
256	512	.777	.713	.517	.600	.759
256	1024	.776	.713	.509	.602	.745
256	2048	.777	.714	.532	.598	.766

n=Samples, ℓ=SeqLen, CMM=CMMLU, MML=MMLU, MBP=MBPP, HE=HumanEval, GSM=GSM8K

Table 6: Qwen2.5-7B-Instruct with G calibration: AWQ (w8a16 Quantization).

n	ℓ	**CMM**	**MML**	**MBP**	**HE**	**GSM**
4	64	.797	.717	.544	.610	.797
4	512	.796	.718	.536	.610	.800
4	1024	.797	.718	.544	.622	.799
4	2048	.797	.717	.558	.610	.811
8	64	.796	.717	.548	.610	.807
8	512	.796	.717	.532	.610	.811
8	1024	.797	.718	.550	.616	.804
8	2048	.797	.717	.554	.604	.809
16	64	.797	.718	.550	.622	.806
16	512	.796	.718	.542	.622	.796
16	1024	.796	.717	.560	.598	.804
16	2048	.797	.718	.542	.604	.801
32	64	.796	.718	.542	.610	.802
32	512	.796	.717	.534	.610	.810
32	1024	.797	.717	.550	.604	.811
32	2048	.796	.717	.540	.634	.807
64	64	.796	.718	.546	.610	.798
64	512	.796	.718	.532	.616	.810
64	1024	.796	.717	.556	.622	.807
64	2048	.796	.717	.556	.604	.809
128	64	.797	.718	.548	.610	.804
128	512	.796	.718	.528	.610	.807
128	1024	.797	.717	.556	.622	.804
128	2048	.796	.717	.554	.604	.806
256	64	.797	.717	.550	.610	.811
256	512	.796	.717	.522	.610	.811
256	1024	.796	.717	.556	.622	.804
256	2048	.796	.717	.552	.604	.811

n=Samples, ℓ=SeqLen, CMM=CMMLU, MML=MMLU, MBP=MBPP, HE=HumanEval, GSM=GSM8K

Table 7: Qwen2.5-7B-Instruct with G calibration: GPTQ (w4a16 Quantization).

n	ℓ	**CMM**	**MML**	**MBP**	**HE**	**GSM**
4	64	.768	.711	.345	.535	.741
4	512	.754	.701	.523	.561	.693
4	1024	.757	.703	.295	.514	.720
4	2048	.751	.705	.331	.516	.716
8	64	.761	.708	.347	.537	.729
8	512	.752	.704	.451	.512	.737
8	1024	.754	.700	.126	.524	.734
8	2048	.752	.707	.328	.549	.718
16	64	.762	.708	.465	.543	.727
16	512	.758	.703	.485	.570	.728
16	1024	.758	.709	.529	.570	.689
16	2048	.753	.709	.327	.555	.740
32	64	.762	.706	.498	.588	.730
32	512	.760	.704	.535	.570	.739
32	1024	.762	.708	.187	.558	.732
32	2048	.764	.711	.228	.552	.743
64	64	.759	.705	.455	.576	.720
64	512	.762	.712	.371	.555	.748
64	1024	.765	.711	.473	.549	.734
64	2048	.766	.710	.254	.581	.742
128	64	.760	.707	.439	.527	.731
128	512	.768	.712	.447	.585	.742
128	1024	.766	.713	.267	.563	.770
128	2048	.764	.713	.238	.561	.742
256	64	.762	.711	.205	.571	.734
256	512	.768	.712	.537	.573	.760
256	1024	.771	.712	.345	.583	.756
256	2048	.767	.714	.203	.567	.744

n=Samples, ℓ=SeqLen, CMM=CMMLU, MML=MMLU, MBP=MBPP, HE=HumanEval, GSM=GSM8K

Table 8: Qwen2.5-7B-Instruct with G calibration: GPTQ (w8a16 Quantization).

n	ℓ	**CMM**	**MML**	**MBP**	**HE**	**GSM**
4	64	0.795	0.717	0.520	0.634	0.805
4	512	0.796	0.717	0.524	0.628	0.809
4	1024	0.796	0.717	0.530	0.616	0.809
4	2048	0.795	0.718	0.538	0.634	0.816
8	64	0.796	0.719	0.514	0.622	0.810
8	512	0.796	0.718	0.518	0.604	0.811
8	1024	0.796	0.717	0.508	0.628	0.796
8	2048	0.796	0.718	0.510	0.640	0.812
16	64	0.796	0.718	0.550	0.622	0.816
16	512	0.795	0.717	0.522	0.604	0.809
16	1024	0.795	0.718	0.512	0.616	0.809
16	2048	0.796	0.718	0.524	0.628	0.812
32	64	0.794	0.719	0.522	0.610	0.813
32	512	0.795	0.718	0.522	0.622	0.826
32	1024	0.794	0.718	0.536	0.622	0.802
32	2048	0.796	0.718	0.528	0.598	0.810
64	64	0.795	0.717	0.512	0.622	0.814
64	512	0.796	0.718	0.508	0.628	0.809
64	1024	0.795	0.718	0.520	0.628	0.817
64	2048	0.794	0.716	0.512	0.640	0.805
128	64	0.795	0.717	0.522	0.634	0.811
128	512	0.795	0.718	0.508	0.616	0.810
128	1024	0.794	0.717	0.528	0.622	0.811
128	2048	0.795	0.717	0.536	0.646	0.809
256	64	0.795	0.717	0.522	0.628	0.805
256	512	0.795	0.718	0.536	0.622	0.811
256	1024	0.795	0.718	0.522	0.604	0.810
256	2048	0.794	0.717	0.522	0.622	0.805

n=Samples, ℓ=SeqLen, CMM=CMMLU, MML=MMLU, MBP=MBPP, HE=HumanEval, GSM=GSM8K

Table 9: Qwen2.5-7B-Instruct with G calibration: SQ+GPTQ(w8a8 Quantization).

n	ℓ	**CMM**	**MML**	**MBP**	**HE**	**GSM**
4	64	.791	.714	.291	.616	.811
4	512	.790	.714	.267	.610	.804
4	1024	.791	.715	.381	.634	.813
4	2048	.789	.715	.294	.614	.801
8	64	.791	.715	.331	.610	.803
8	512	.790	.717	.341	.642	.802
8	1024	.792	.714	.356	.606	.806
8	2048	.789	.715	.291	.626	.801
16	64	.790	.715	.388	.608	.812
16	512	.791	.716	.275	.630	.805
16	1024	.790	.715	.235	.589	.810
16	2048	.791	.714	.262	.620	.808
32	64	.790	.714	.355	.604	.808
32	512	.791	.714	.317	.638	.808
32	1024	.790	.716	.372	.636	.810
32	2048	.791	.714	.245	.620	.813
64	64	.791	.714	.285	.628	.802
64	512	.790	.714	.281	.630	.812
64	1024	.790	.715	.301	.626	.806
64	2048	.790	.713	.265	.600	.813
128	64	.791	.715	.330	.602	.801
128	512	.790	.713	.406	.614	.816
128	1024	.792	.714	.247	.620	.808
128	2048	.790	.713	.240	.618	.801
256	64	.790	.716	.353	.628	.806
256	512	.792	.713	.381	.628	.803
256	1024	.790	.713	.311	.636	.802
256	2048	.790	.713	.354	.650	.811

n=Samples, ℓ=SeqLen, CMM=CMMLU, MML=MMLU, MBP=MBPP, HE=HumanEval, GSM=GSM8K

A.2 Quantization Results for Llama-3.1-8B-Instruct

Table 10: Llama-3.1-8B-Instruct w4a16 AWQ results.

n	ℓ	CMM	MML	MBP	HE	GSM
4	64	.513	.628	.515	.567	.717
4	512	.515	.629	.510	.575	.698
4	1024	.514	.629	.514	.555	.712
4	2048	.514	.629	.515	.549	.691
8	64	.511	.623	.521	.539	.728
8	512	.512	.628	.516	.551	.707
8	1024	.515	.633	.504	.537	.724
8	2048	.515	.631	.514	.567	.698
16	64	.513	.624	.517	.541	.726
16	512	.512	.629	.515	.553	.702
16	1024	.514	.633	.520	.555	.718
16	2048	.514	.632	.515	.563	.697
32	64	.513	.624	.517	.541	.726
32	512	.513	.628	.514	.553	.704
32	1024	.514	.630	.510	.551	.694
32	2048	.515	.632	.514	.565	.696
64	64	.513	.624	.517	.541	.726
64	512	.513	.628	.515	.557	.699
64	1024	.514	.630	.510	.549	.689
64	2048	.514	.632	.514	.565	.698
128	64	.514	.624	.517	.541	.725
128	512	.515	.629	.513	.579	.698
128	1024	.514	.629	.511	.573	.699
128	2048	.514	.632	.517	.567	.695
256	64	.511	.624	.513	.549	.732
256	512	.514	.629	.509	.577	.698
256	1024	.515	.629	.509	.575	.696
256	2048	.514	.632	.516	.569	.696

n=Samples, ℓ=SeqLen, CMM=CMMLU, MML=MMLU, MBP=MBPP, HE=HumanEval, GSM=GSM8K

Table 11: Llama-3.1-8B-Instruct w8a16 AWQ results.

n	ℓ	CMM	MML	MBP	HE	GSM
4	64	.536	.631	.590	.585	.789
4	512	.537	.632	.590	.585	.785
4	1024	.536	.631	.602	.573	.793
4	2048	.537	.630	.586	.585	.789
8	64	.537	.631	.590	.585	.793
8	512	.537	.632	.598	.592	.787
8	1024	.536	.632	.594	.598	.793
8	2048	.537	.632	.590	.585	.797
16	64	.535	.630	.596	.573	.790
16	512	.536	.632	.602	.585	.791
16	1024	.536	.633	.594	.598	.801
16	2048	.536	.632	.596	.598	.797
32	64	.536	.630	.588	.592	.798
32	512	.536	.632	.602	.585	.791
32	1024	.536	.631	.596	.573	.792
32	2048	.537	.631	.598	.579	.793
64	64	.537	.632	.590	.592	.789
64	512	.537	.632	.602	.585	.789
64	1024	.536	.632	.594	.598	.790
64	2048	.536	.632	.588	.598	.798
128	64	.536	.631	.592	.573	.792
128	512	.536	.632	.602	.585	.792
128	1024	.536	.632	.594	.585	.789
128	2048	.536	.632	.588	.598	.795
256	64	.536	.631	.594	.573	.792
256	512	.537	.632	.602	.585	.790
256	1024	.535	.631	.590	.585	.792
256	2048	.536	.631	.604	.585	.798

n=Samples, ℓ=SeqLen, CMM=CMMLU, MML=MMLU, MBP=MBPP, HE=HumanEval, GSM=GSM8K

Table 12: Llama-3.1-8B-Instruct with G calibration: GPTQ w4a16. Note: coding benchmarks (HumanEval, MBPP) exhibit severe performance collapse under certain calibration settings.

n	ℓ	**CMM**	**MML**	**MBP**	**HE**	**GSM**
4	64	.507	.600	.004	.073	.137
4	512	.473	.604	.002	.085	.128
4	1024	.471	.592	.004	.055	.131
4	2048	.464	.576	.000	.043	.137
8	64	.484	.602	.000	.085	.135
8	512	.468	.597	.004	.073	.138
8	1024	.490	.592	.004	.079	.155
8	2048	.485	.602	.004	.055	.150
16	64	.486	.600	.006	.063	.123
16	512	.484	.593	.001	.061	.133
16	1024	.486	.608	.003	.057	.140
16	2048	.486	.600	.001	.077	.152
32	64	.484	.602	.001	.077	.146
32	512	.487	.597	.005	.067	.132
32	1024	.491	.610	.001	.073	.180
32	2048	.484	.610	.003	.073	.144
64	64	.488	.607	.001	.073	.162
64	512	.483	.598	.012	.061	.178
64	1024	.491	.609	.005	.091	.203
64	2048	.479	.604	.078	.079	.171
128	64	.486	.591	.001	.059	.155
128	512	.492	.616	.247	.106	.237
128	1024	.495	.613	.407	.114	.289
128	2048	.484	.601	.475	.098	.251
256	64	.481	.586	.001	.045	.133
256	512	.494	.615	.004	.077	.234
256	1024	.498	.614	.022	.067	.239
256	2048	.501	.615	.004	.075	.198

n=Samples, ℓ=SeqLen, CMM=CMMLU, MML=MMLU, MBP=MBPP, HE=HumanEval, GSM=GSM8K

Table 13: Llama-3.1-8B-Instruct w8a16 GPTQ results.

n	ℓ	**CMM**	**MML**	**MBP**	**HE**	**GSM**
4	64	.536	.632	.581	.591	.790
4	512	.538	.631	.589	.585	.783
4	1024	.535	.632	.582	.593	.791
4	2048	.536	.633	.588	.591	.791
8	64	.535	.633	.579	.593	.792
8	512	.537	.632	.583	.593	.790
8	1024	.537	.633	.586	.591	.782
8	2048	.536	.631	.578	.593	.792
16	64	.536	.633	.577	.604	.790
16	512	.536	.632	.577	.589	.790
16	1024	.537	.631	.582	.598	.794
16	2048	.536	.632	.582	.598	.784
32	64	.537	.633	.578	.591	.789
32	512	.535	.631	.583	.589	.788
32	1024	.536	.632	.589	.591	.786
32	2048	.538	.633	.589	.602	.795
64	64	.537	.630	.585	.593	.783
64	512	.537	.631	.584	.604	.788
64	1024	.537	.632	.583	.602	.794
64	2048	.539	.632	.580	.593	.787
128	64	.537	.631	.584	.575	.783
128	512	.534	.633	.583	.583	.794
128	1024	.535	.633	.583	.589	.794
128	2048	.536	.632	.585	.598	.784
256	64	.537	.633	.573	.593	.774
256	512	.536	.631	.580	.587	.789
256	1024	.535	.632	.587	.606	.797
256	2048	.537	.632	.585	.604	.794

n=Samples, ℓ=SeqLen, CMM=CMMLU, MML=MMLU, MBP=MBPP, HE=HumanEval, GSM=GSM8K

Table 14: Llama-3.1-8B-Instruct with G calibration: SQ+GPTQ(w8a8 Quantization).

n	ℓ	**CMM**	**MML**	**MBP**	**HE**	**GSM**
4	64	.530	.633	.569	.569	.767
4	512	.529	.633	.572	.559	.770
4	1024	.532	.635	.570	.545	.776
4	2048	.531	.631	.554	.543	.766
8	64	.533	.634	.557	.559	.767
8	512	.532	.633	.570	.563	.763
8	1024	.532	.631	.565	.555	.767
8	2048	.533	.635	.571	.577	.765
16	64	.532	.634	.565	.571	.766
16	512	.531	.634	.575	.567	.763
16	1024	.533	.635	.577	.567	.774
16	2048	.531	.635	.568	.547	.763
32	64	.533	.635	.573	.559	.779
32	512	.531	.635	.574	.567	.765
32	1024	.531	.636	.579	.567	.770
32	2048	.533	.632	.570	.559	.771
64	64	.530	.631	.564	.573	.773
64	512	.531	.634	.569	.569	.772
64	1024	.531	.636	.575	.561	.770
64	2048	.533	.629	.559	.561	.775
128	64	.535	.634	.571	.557	.770
128	512	.532	.633	.560	.557	.776
128	1024	.528	.635	.564	.573	.771
128	2048	.530	.629	.573	.559	.775
256	64	.532	.635	.569	.567	.770
256	512	.533	.632	.579	.561	.766
256	1024	.533	.634	.571	.555	.768
256	2048	.530	.632	.561	.561	.773

n=Samples, ℓ=SeqLen, CMM=CMMLU, MML=MMLU, MBP=MBPP, HE=HumanEval, GSM=GSM8K