Fisher-Guided Cross-Layer Budget Rebalancing for Mixed-Precision MoE Quantization

Anonymous ACL submission

Abstract

Mixed-precision quantization allocates different bitwidths to different model components, but for Mixture-of-Experts (MoE) architectures with hundreds to thousands of expert sub-networks, a fundamental question remains underexplored: *where* should the bit budget go—within layers or across layers? We study this question on the Qwen-MoE family under an RTN quantization pipeline applied to expert weights. Our central finding is that cross-layer budget rebalancing—shifting bits from less-sensitive to more-sensitive layers using a lightweight z-score heuristic—substantially outperforms per-layer optimization, even when the per-layer allocation uses the same Fisher sensitivity ranking. On Qwen2-57B-A14B, cross-layer rebalancing improves WikiText-2 perplexity by 2.15 over per-layer Fisher bucket allocation at 3.0-bit average expert budget, while achieving the best downstream accuracy on ARC-Challenge, WinoGrande, HellaSwag, BoolQ, SciQ, and OpenBookQA. We further observe that perplexity improvements do not always translate into downstream accuracy gains—activation-frequency and uniform 3-bit allocation achieve the lowest PPL on the 57B model yet underperform Fisher-based methods on benchmarks—and that the preferred rebalancing strength depends on model scale. Our results suggest that future MoE quantization work should prioritize cross-layer allocation decisions alongside per-expert sensitivity estimation.

1 Introduction

Mixture-of-Experts (MoE) architectures (Shazeer et al., 2017; Fedus et al., 2022) are an important paradigm for scaling large language models, including systems such as Mixtral (Jiang et al., 2024), DeepSeek-V3 (DeepSeek-AI, 2024), and Qwen-MoE (Yang et al., 2024). By activating only a subset of experts per token, MoE models increase parameter capacity without activating all parameters on every forward pass. However, their large total parameter counts—often exceeding tens of billions—make weight storage and deployment on resource-constrained hardware challenging. Post-training quantization (PTQ) (Frantar et al., 2022; Lin et al., 2025; Xiao et al., 2023) compresses model weights to low bitwidths without retraining. In this work, we deliberately study an isolated setting: per-channel round-to-nearest (RTN) quantization applied strictly to MoE expert weights. Our core contribution is the *cross-layer budget allocation strategy*, not the quantization operator itself. While advanced quantizers (e.g., GPTQ, AWQ) yield better absolute performance, their MoE implementations entangle operator-level optimizations with system constraints, masking pure allocation effects. Using a fundamental RTN quantizer creates an interference-free environment to validate Fisher-guided allocation. We ask: *given a basic quantizer, can expert-sensitivity-aware cross-layer allocation significantly outperform standard baselines?*

The key insight of this work is that not all experts contribute equally: different experts and transformer layers exhibit vastly different sensitivity to quantization. This motivates *mixed-precision quantization*, where sensitive experts receive higher bitwidths and robust experts are aggressively compressed. Two critical questions arise: (1) how to measure expert sensitivity accurately, and (2) how to allocate a global bitwidth budget across layers and experts. Most prior mixed-precision work focuses on sensitivity metrics while treating budget allocation as a simple per-layer partition, leaving open a fundamental question: *given a fixed global bit budget, how much does cross-layer allocation matter?*

We study this question through a Fisher trace-guided framework: we formalize expert-level mixed-precision quantization as a two-level resource allocation problem and propose a cross-

layer budget rebalancing heuristic. Our contributions are:

1. **A Fisher-guided two-level allocation framework for MoE quantization.** We formalize expert-level mixed-precision as a two-level Multiple-Choice Knapsack Problem (MCKP), propose Fisher trace scoring for expert sensitivity, and derive a lightweight z-score cross-layer rebalancing heuristic from the DP marginal-benefit principle, controlled by a single hyperparameter α.

2. **Cross-layer allocation dominates within-layer optimization.** Under the same Fisher ranking, cross-layer rebalancing improves WikiText-2 PPL by 0.39 on 2.7B and 2.15 on 57B over per-layer Fisher Bucket, winning on five of six downstream benchmarks on the 2.7B model.

3. **Fisher trace as an effective expert sensitivity signal.** We systematically compare Fisher trace with GradNorm and activation frequency. Fisher trace outperforms both on PPL and downstream benchmarks, capturing curvature rather than just gradient magnitude or usage frequency.

4. **Practical observations from two-model study across six benchmarks.** On Qwen1.5-MoE-2.7B and Qwen2-57B-A14B, we find that the preferred rebalancing strength α depends on model scale, and that perplexity improvements do not always translate into downstream accuracy gains—activation frequency and uniform 3-bit allocation achieve the lowest 57B PPL yet underperform Fisher-based methods on benchmarks.

2 Related Work

2.1 MoE Architectures

MoE models (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022; Zoph, 2022) route each token to a subset of expert feed-forward networks. Related work studies expert specialization and routing (Zhang et al., 2022; Chen et al., 2024). Modern MoE LLMs (Jiang et al., 2024; DeepSeek-AI, 2024; Yang et al., 2024) activate 4–8 out of 60–256 experts per layer. We study Qwen1.5-MoE-A2.7B (24 layers, 60 experts, ∼12.5B expert params) and Qwen2-57B-A14B (28 layers, 64 experts, ∼50B expert params).

2.2 Post-Training Quantization

PTQ methods compress pre-trained models without costly retraining. GPTQ (Frantar et al., 2022) uses approximate second-order information for layer-wise quantization; AWQ (Lin et al., 2025) identifies salient weight channels via activation statistics; SmoothQuant (Xiao et al., 2023) migrates quantization difficulty from activations to weights. Other recent LLM compression methods improve low-bit deployment through outlier-aware matrix multiplication (Dettmers et al., 2022), quantized adaptation (Dettmers et al., 2023), incoherence processing and codebook design (Tseng et al., 2024), or sparsity–quantization combinations (Frantar and Alistarh, 2023). These methods target dense transformers with uniform bitwidth allocation. We omit GPTQ/AWQ comparisons because current public tooling does not yet provide stable support for Qwen2-MoE expert-level mixed precision.

2.3 Budget Allocation for Mixed-Precision Quantization

Mixed-precision quantization assigns different bitwidths to different parts of a model. HAWQ (Dong et al., 2019, 2020) uses Hessian spectrum and trace information for mixed-precision CNN quantization, while ZeroQuant (Yao et al., 2022) studies affordable PTQ for large transformers. For LLMs, recent sparsity work analyzes how pruning or sparsity scales in large language models (Hossain et al., 2026). Closest to our setting, emerging MoE-specific quantization work studies expert-aware compression and theoretical guarantees (Chowdhury et al., 2026), and modality-adaptive quantization has been explored for MoE vision-language models (Qin et al., 2026). In contrast to methods that solve formal knapsack-like optimization problems, our cross-layer rebalancing uses a lightweight z-score heuristic that requires no iterative solver and is controlled by a single hyperparameter.

3 Method

3.1 Overview

Given a pre-trained MoE model with L layers and E experts per layer, and a global average bitwidth budget B, our goal is to assign a bitwidth $b_{\ell,e} \in \{2, 3, 4\}$ to each expert e in layer ℓ such that the

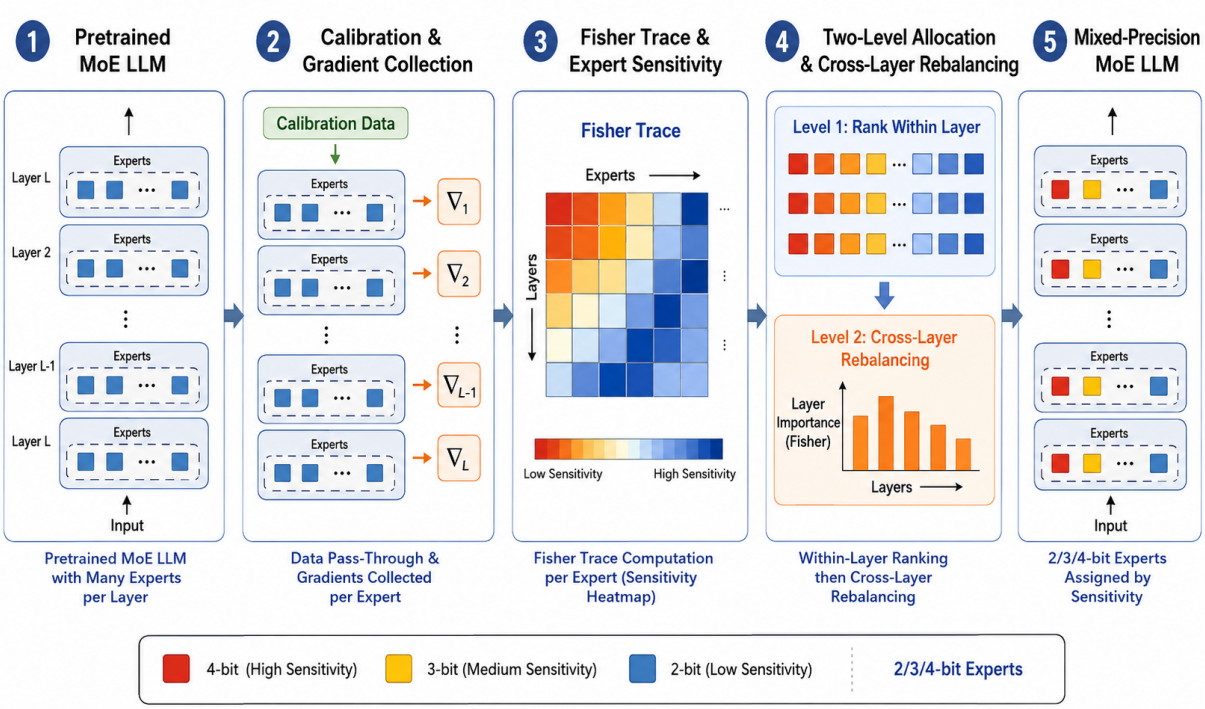

Figure 1: Overview of our Fisher-guided mixed-precision quantization framework. We estimate expert-level Fisher sensitivity from calibration gradients, construct a layer-by-expert sensitivity matrix, and allocate bitwidths through within-layer expert ranking and cross-layer budget rebalancing. Sensitive experts receive higher precision while robust experts are compressed more aggressively.

average bitwidth equals B and the quantization-induced performance degradation is minimized.

This is a two-level resource allocation problem. At the lower level, we select which experts receive 2, 3, or 4 bits given a per-layer budget b_ℓ; at the upper level, we distribute the global budget B into per-layer budgets $\{b_\ell\}_{\ell=1}^{L}$. A layer's sensitivity to its budget depends on the composition of its experts.

The lower-level problem is a Multiple-Choice Knapsack Problem (MCKP): each expert is an item with three bitwidth choices, each with a cost (the bitwidth) and value (negative quantization error). When experts within a layer have equal parameter counts—as in Qwen-MoE—the MCKP reduces to ranking experts by value and assigning higher bitwidths to top-ranked experts (the bucket strategy).

The upper-level cross-layer allocation is harder. Solving it as a nested MCKP via dynamic programming (DP) consistently allocates more bits to layers with higher Fisher trace, approximately equalizing Fisher-weighted marginal benefit. This motivates a lightweight z-score heuristic: shift budget toward layers with above-average Fisher importance (visualized in Figure 2).

We apply RTN quantization with per-channel scaling to each expert's weight matrices:

$$\hat{W} = \text{clamp}\Big(\text{round}\big(\frac{W}{s}\big),\ -2^{b-1},\ 2^{b-1}-1\Big)\cdot s \tag{1}$$

where $s = \max(|W|, \text{dim} = -1)\,/\,(2^{b-1}-1)$ is computed per output channel. It applies to each expert's feed-forward weight matrices only (up-projection, down-projection, and gate projections); attention, embedding, and normalization weights remain in bf16.

3.2 Fisher Trace for Expert Sensitivity

For each expert block, we compute a Fisher trace score as a proxy for its RTN quantization sensitivity. Given calibration data $\mathcal{D}$ and expert parameter set $\mathcal{P}_{\ell,e}$, the empirical diagonal Fisher score used by our method is

$$s_{\ell,e}^{\text{Fisher}} = \sum_{i\in\mathcal{P}_{\ell,e}} \frac{1}{K}\sum_{k=1}^{K}\left(\frac{\partial \log p(x_k \mid \theta)}{\partial \theta_i}\right)^2. \tag{2}$$

This is the unnormalized trace of the expert-specific empirical Fisher submatrix. It is inten-

tionally *not* divided by the number of parameters in the expert, because our allocation problem concerns the total expected loss perturbation contributed by the whole expert block. In implementation, Eq. 2 is obtained by accumulating squared gradients over calibration batches and summing over all weight tensors belonging to the expert.

To avoid a possible ambiguity, we explicitly distinguish this score from the Gradient Norm baseline. The Gradient Norm baseline computes a parameter-count-normalized first-order magnitude,

$$s_{\ell,e}^{\mathrm{GradNorm}} = \frac{1}{|\mathcal{P}_{\ell,e}|}\left(\sum_{i\in\mathcal{P}_{\ell,e}} |\bar{g}_i|^2\right)^{1/2}, \qquad \bar{g}_i = \frac{1}{K}\sum_{k=1}^{K}\frac{\partial \log p(x_k \mid \theta)}{\partial \theta_i}. \tag{3}$$

followed by the same per-layer ranking and bucket allocation procedure. Thus, the two baselines differ both mathematically and operationally: Fisher sums expected squared per-sample gradients as a trace-like curvature proxy, whereas GradNorm uses a normalized gradient-magnitude statistic. The comparison therefore tests the sensitivity metric, not merely a different normalization of the same quantity. Experts with higher Fisher trace are preferentially assigned higher precision.

3.3 Two-Level Budget Allocation

3.3.1 Per-Layer Bucket Allocation (Baseline)

Given sensitivity scores $\{s_{\ell,e}\}_{e=1}^{E}$ for layer ℓ and per-layer budget b_ℓ, we rank experts by sensitivity and assign bitwidths via a simple "bucket" strategy: the top-ranked experts receive 4-bit, the bottom receive 2-bit, and the middle receive 3-bit, chosen to match the target average bitwidth:

$$\begin{aligned} w_4 &= \left\lfloor \frac{(b_\ell - 2)E}{2} \right\rfloor, \\ w_3 &= \mathrm{round}\big((b_\ell - 2)E - 2w_4\big), \\ w_2 &= E - w_4 - w_3. \end{aligned} \tag{4}$$

When all layers receive the same budget $b_\ell = B$, this is the standard **Fisher Bucket** baseline. When b_ℓ is near the extremes, w_3 may fall outside $[0, E-w_4]$; in those cases we clamp w_3 to the valid range and absorb the difference into w_2.

3.3.2 Cross-Layer Rebalancing (Ours)

The key limitation of the bucket approach is the assumption that all layers deserve equal budgets ($b_\ell = B$). In practice, earlier layers are often more sensitive to quantization than later layers (Yao et al., 2022), as confirmed by the Fisher trace heatmap in Figure 2. This raises the upper-level allocation question: given L layers with distinct sensitivity profiles, how should B be partitioned?

Formally, the cross-layer problem is another MCKP where each layer is a "meta-item" whose value depends on its budget. Our DP analysis shows a consistent pattern: more bits go to layers with higher Fisher trace, approximately equalizing the Fisher-weighted marginal benefit $\bar{s}_\ell \cdot q'(b_\ell)$, where $q'(\cdot)$ denotes the marginal reduction in quantization error. We therefore use per-layer Fisher importance $\bar{s}_\ell = \frac{1}{E}\sum_e s_{\ell,e}$ as a proxy for the marginal benefit of assigning budget to layer ℓ.

We operationalize this intuition through a z-score–based rebalancing formula:

$$b_\ell = B + \alpha \cdot \frac{\log \bar{s}_\ell - \mu}{\sigma}, \quad \bar{s}_\ell = \frac{1}{E}\sum_{e=1}^{E} s_{\ell,e} \tag{5}$$

where μ and σ are the mean and standard deviation of $\log \bar{s}_\ell$ across layers, and α controls the rebalancing strength. Per-layer budgets are clipped to $[2.0, 4.0]$; when clipping shifts the mean away from B, we apply a uniform additive correction to unclipped budgets and repeat until convergence. In practice, Fisher traces are strictly positive (squared gradients summed over thousands of parameters), so the log-transform is well-defined.

This formulation has a natural interpretation: $\alpha = 0$ recovers the uniform bucket baseline; $\alpha > 0$ shifts bits from low-importance to high-importance layers in proportion to their deviation from the mean. The log-transform compresses the dynamic range of Fisher traces, preventing a few outlier layers from dominating the allocation. The resulting per-layer budgets are then used as input to the per-layer bucket allocation described above.

The z-score heuristic therefore approximates the DP marginal-benefit principle with a single linear rebalancing operation. It has one tunable hyperparameter α, requires no iterative optimization, and adds negligible overhead.

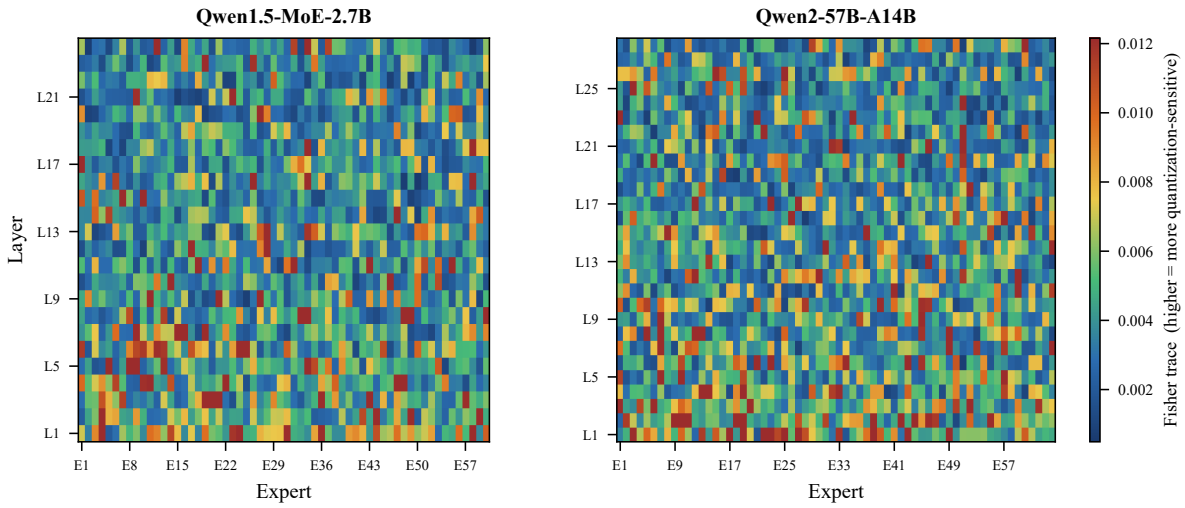

Figure 2: Fisher trace sensitivity matrices for Qwen1.5-MoE-2.7B @2.8b (left) and Qwen2-57B @3.0b (right). Rows are layers, columns are experts. Darker cells indicate higher Fisher trace. Early layers are consistently more sensitive across both model scales, and a small fraction of experts dominate the sensitivity within each layer, motivating expert-level mixed-precision allocation.

4 Experiments

4.1 Experimental Setup

Models. We evaluate on two Qwen-MoE models: Qwen1.5-MoE-A2.7B (24 layers, 60 experts/layer, ∼12.5B expert parameters) and Qwen2-57B-A14B-Instruct (28 layers, 64 experts/layer, ∼50B expert parameters). The original checkpoints are loaded in bf16; quantized evaluation uses the simulated-quantization procedure described in Section 3.3.

Calibration. Fisher traces are computed using 20–50 batches of WikiText-2 with sequence length 2048. PPL is evaluated on the WikiText-2 test set.

Baselines and evaluation. We compare allocation strategies under the same RTN simulated-quantization implementation: (1) **BF16** (unquantized) as an upper bound; (2) **Uniform W3 (expert)** applying 3-bit RTN uniformly to all experts; (3) **GradNorm**-based sensitivity ranking; (4) **Activation Frequency**-based ranking; (5) **Fisher Bucket** with a uniform per-layer budget; and (6) **Fisher Rebalanced**, our cross-layer allocation method. Downstream tasks are evaluated with the language-model evaluation protocol (Gao et al., 2021) and include ARC-Challenge (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2021), HellaSwag (Zellers et al., 2019), BoolQ (Clark et al., 2019), SciQ (Welbl et al., 2017), and OpenBookQA (Mihaylov et al., 2018).

Hardware. All experiments run on 4×NVIDIA A800-80GB GPUs. The 2.7B model uses a single GPU; Each PPL evaluation takes ∼1–3 minutes (2.7B) or ∼3 minutes (57B). Full benchmark evaluation takes ∼3 hours per model (6 benchmarks × 3 configurations).

Method	2.7B @2.8b	57B @3.0b
BF16 (baseline)	6.79	39.00
Uniform W3 (expert)	8.54	44.08
GradNorm Bucket	9.19	47.17
Activation Freq	9.27	44.09
Fisher Bucket	8.89	46.74
Fisher Rebalanced (ours)	**8.50**	**44.59**

Table 1: WikiText-2 perplexity (lower is better). All methods except BF16 apply RTN quantization to expert feed-forward weights only (attention, embedding, and normalization layers remain in bf16).

4.2 Perplexity Results

Table 1 reports WikiText-2 PPL. Under our expert-only RTN pipeline, uniform 3-bit quantization of all experts yields a moderate degradation on 2.7B (PPL 8.54 vs. BF16 6.79), but mixed-precision allocation recovers substantial performance. Fisher-based methods outperform GradNorm allocation on both models, and cross-layer rebalancing further improves over per-layer Fisher Bucket.

Our Fisher Rebalanced method achieves the best PPL on the 2.7B model (8.50, 0.39 improvement over Fisher Bucket). On the 57B model, Fisher Rebalanced achieves 44.59 (2.15 improve-

ment over Fisher Bucket), though uniform 3-bit (44.08) and activation frequency (44.09) achieve slightly lower absolute PPL in this setting. Activation frequency's competitive PPL is consistent with prior findings that larger MoE models are more activation-dominated. As we show in Section 4.3, however, Fisher-based methods remain stronger on downstream benchmarks.

4.3 Downstream Benchmark Results

Interpreting downstream absolute scores. Some absolute accuracies, especially HellaSwag for Qwen2-57B-A14B, are lower than common leaderboard numbers because we use one shared raw language-model evaluation protocol rather than model-specific chat templates, prompt wrappers, or answer-format tuning. Since all quantization methods use the same prompt format, shot setting, decoding protocol, and scoring implementation, the relative comparisons in Tables 2–3 remain internally controlled. Similarly, the elevated BF16 WikiText-2 PPL of 39.00 on the 57B model (vs. 6.79 on 2.7B) reflects the instruction-tuned nature of Qwen2-57B-A14B-Instruct, and the corresponding HellaSwag score of 26.2% is near the 25% random-chance baseline for 4-way multiple choice. Relative comparisons remain meaningful under the shared evaluation protocol.

To examine whether PPL improvements are reflected beyond language modeling loss, we evaluate on six diverse benchmarks: ARC-Challenge (Clark et al., 2018) (science QA, 25-shot), WinoGrande (Sakaguchi et al., 2021) (commonsense reasoning, 5-shot), HellaSwag (Zellers et al., 2019) (commonsense NLI, 10-shot), BoolQ (Clark et al., 2019) (reading comprehension, 5-shot), SciQ (Welbl et al., 2017) (science QA, 5-shot), and OpenBookQA (Mihaylov et al., 2018) (science QA, 5-shot).

Table 2 presents the downstream benchmark results. Fisher Rebalanced wins on 5 of 6 benchmarks, with gains over Fisher Bucket ranging from +1.0% (OpenBookQA) to +5.6% (BoolQ). The key pattern is that cross-layer allocation can dominate the choice of sensitivity metric: although GradNorm Bucket beats Fisher Bucket on BoolQ, HellaSwag, and OpenBookQA, Fisher Rebalanced uses the same Fisher ranking as Fisher Bucket and still wins on five of six benchmarks. Uniform W3 is also competitive and wins OpenBookQA, indicating that expert-only uniform quantization is a strong baseline.

On the 57B model (Table 3), Fisher Rebalanced wins or ties on all six benchmarks. OpenBookQA shows the clearest effect: +6.0% over Fisher Bucket with identical per-expert Fisher scores, so the gain comes entirely from cross-layer rebalancing. GradNorm nearly matches Fisher Bucket on five tasks but drops sharply on SciQ (64.3% vs. 77.2%), revealing brittleness in that signal. Activation Frequency achieves better PPL than Fisher Bucket and Fisher Rebalanced, yet Fisher-based methods remain stronger on downstream benchmarks, underscoring the need for task-level evaluation.

4.4 Ablation Studies

4.4.1 Rebalancing Strength α

Table 4 shows the effect of α on PPL. Optimal rebalancing strength increases with model scale: α=0.2 for 2.7B, α=0.4 for 57B. Larger α (0.6) degrades performance by over-allocating bits to a few layers and starving others.

4.4.2 Compression Analysis

Table 5 shows the theoretical compression ratios. Our mixed-precision approach yields 81–84% estimated memory reduction for expert parameters compared to bf16, while using lower average bitwidth than uniform 4-bit storage.

5 Analysis

5.1 Why Does the Rebalancing Heuristic Work?

The z-score heuristic approximates the DP-based MCKP solver, which allocates more bits to high-Fisher layers to equalize marginal benefit. The heuristic captures this by shifting budget in proportion to standardized Fisher trace.

This works because per-layer sensitivity is far from uniform: early layers (0–6) exhibit substantially higher Fisher trace than later layers (20–27), so a uniform budget under-allocates them. The log-transform in Eq. 5 further prevents a few outlier layers from dominating the budget, avoiding the starvation effect observed at high α values (Table 4).

5.2 Scale-Dependent Phenomena

Within the Qwen-MoE family studied here, we observe two scale-dependent effects. First, the preferred α increases with model scale (α=0.2 for

Benchmark	BF16	Uniform W3 (expert)	GradNorm Bucket	ActFreq	F.Bucket	F.Rebal (ours)
ARC-Challenge	76.5	67.1	64.1	61.3	65.4	**68.3**
WinoGrande	54.9	53.7	51.5	51.9	51.9	**52.7**
HellaSwag	44.3	34.6	33.7	31.4	32.7	**36.7**
BoolQ	80.6	75.7	75.8	71.7	71.2	**76.8**
SciQ	88.1	81.6	78.5	78.5	82.7	**85.4**
OpenBookQA	73.2	**62.8**	61.8	58.6	59.4	60.4

Table 2: Qwen1.5-MoE-2.7B benchmark accuracy @2.8b average expert bitwidth. All methods share the same RTN simulated-quantization pipeline and evaluation protocol. Bold indicates best among quantized methods.

Benchmark	BF16	GradNorm Bucket	ActFreq	F.Bucket	F.Rebal (ours)
ARC-Challenge	70.2	65.7	65.3	65.7	**66.9**
WinoGrande	52.1	52.1	52.3	52.1	**52.6**
HellaSwag	26.2	25.2	25.2	25.2	**25.3**
BoolQ	74.1	71.0	71.1	71.0	**71.0**
SciQ	79.2	64.3	74.9	77.2	**77.5**
OpenBookQA	58.0	49.6	49.6	49.6	**55.6**

Table 3: Qwen2-57B-A14B benchmark accuracy @3.0b budget (α=0.4). All methods share the same RTN pipeline. Bold indicates best among quantized methods.

Configuration	α=0.2	α=0.4	α=0.6
2.7B @2.5b	**10.32**	10.96	11.48
2.7B @2.8b	**8.50**	8.58	8.83
57B @3.0b	45.99	**44.59**	45.40

Table 4: Effect of rebalancing strength α on PPL. Bold indicates best per budget. The optimal α shifts from 0.2 (2.7B) to 0.4 (57B), suggesting larger models benefit from more aggressive cross-layer rebalancing.

Configuration	Avg bits	Expert GB	Saving
Qwen1.5-MoE-2.7B (12.5B expert params)			
BF16	16.0	24.9	—
Uniform W4 (theor.)	4.0	6.2	75%
Fisher Rebal @2.8b	2.8	4.4	**82%**
Qwen2-57B-A14B (∼50B expert params)			
BF16	16.0	100.0	—
Uniform W4 (theor.)	4.0	25.0	75%
Fisher Rebal @3.0b	3.0	18.8	**81%**

Table 5: Theoretical memory compression analysis for expert parameters under physical INT storage assumptions. Savings are based on the assigned INT2/INT3/INT4 expert bitwidths and exclude packing overheads and non-expert parameters. "Uniform W4 (theor.)" shows the storage savings if all experts were stored at 4-bit precision. Evaluation itself uses simulated quantization with bf16 dequantized tensors; the savings reported here are theoretical estimates.

2.7B, α=0.4 for 57B), suggesting that larger MoE models exhibit greater inter-layer sensitivity variance. Second, activation frequency becomes more PPL-competitive on the larger model (44.09 vs. 44.59), consistent with skewed routing patterns, yet Fisher-based allocation remains stronger on downstream benchmarks. We treat these as empirical observations on the Qwen-MoE family rather than claims about MoE architectures in general.

5.3 The PPL vs. Downstream Accuracy Disconnect

A notable finding in our 57B results is the divergence between language modeling perplexity and downstream task accuracy. While Activation Frequency achieves a lower WikiText-2 PPL (44.09) than Fisher Rebalanced (44.59), it consistently underperforms on knowledge-intensive downstream tasks like SciQ and OpenBookQA. This paradox underscores the importance of evaluating quantization methods on end tasks rather than relying solely on language modeling loss.

This disconnect stems from the differing token distributions evaluated by PPL versus downstream QA. Language modeling PPL is heavily dominated by frequent, functional tokens. Consequently, activation frequency allocation assigns higher precision to frequently activated "syntax-heavy" experts, while aggressively compressing sparsely activated "knowledge-heavy" experts—which handle rare, specialized entities crucial for QA—down to 2-bit.

In contrast, Fisher trace captures the *curvature*

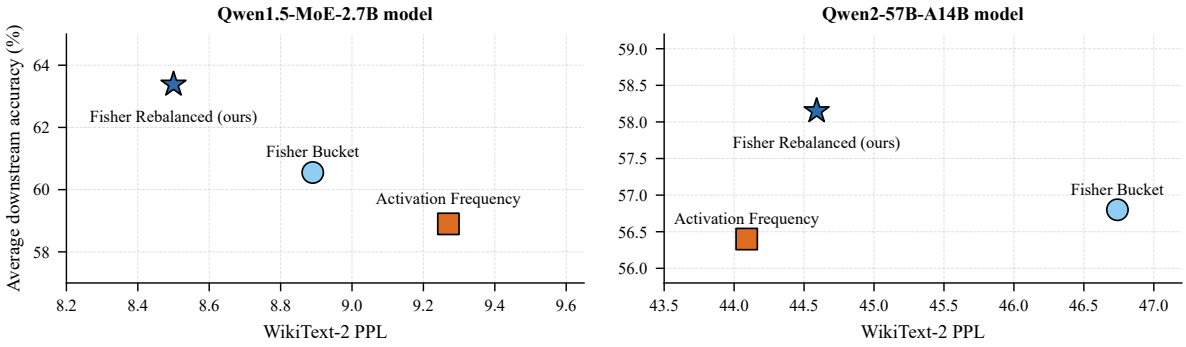

Figure 3: Perplexity–accuracy trade-off (Left: 2.7B @2.8b; Right: 57B @3.0b). Points show WikiText-2 PPL versus average accuracy across six downstream benchmarks. Fisher Rebalanced (blue star) outperforms Fisher Bucket on both metrics. Activation Frequency (orange) achieves competitive PPL but lower accuracy, confirming PPL is an imperfect proxy for task performance.

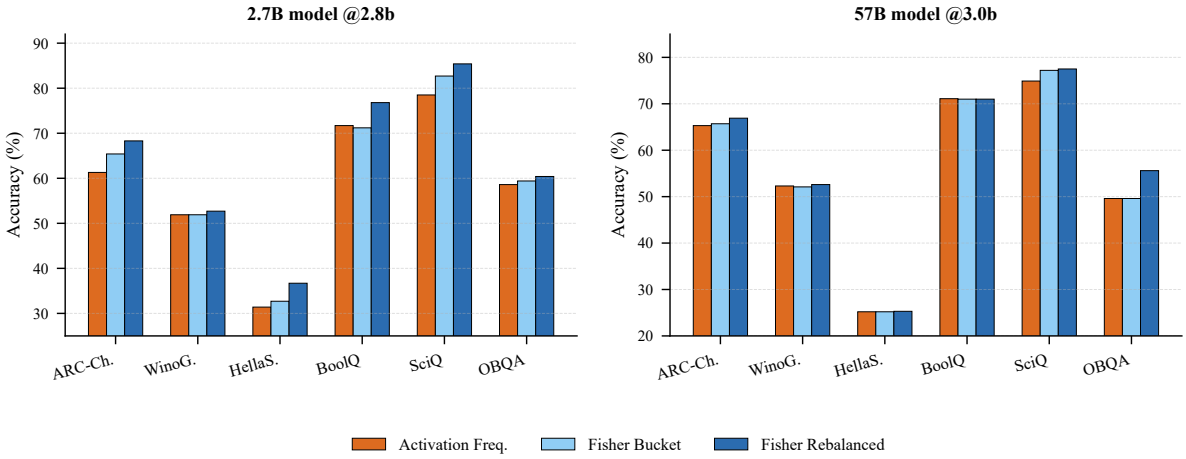

Figure 4: Per-benchmark accuracy on 2.7B @2.8b (left) and 57B @3.0b (right). Fisher Rebalanced (blue) generally outperforms Fisher Bucket (light blue) and Activation Frequency (orange) across tasks. See Section 4.1 for caveats regarding absolute scores.

of the loss landscape. If a knowledge-heavy expert is severely quantized, the prediction for a critical factual token degrades, inducing a massive gradient spike during calibration. Fisher therefore assigns high sensitivity to these critical experts, protecting them from over-compression regardless of their low activation frequency. This highlights a crucial insight for MoE compression: optimizing purely for average token loss (PPL) structurally biases against rare-knowledge experts, whereas gradient-based metrics successfully decouple an expert's functional importance from its raw activation count.

6 Conclusion

We presented a Fisher-guided cross-layer mixed-precision allocation framework for RTN quantization of Qwen-MoE expert weights. Our method addresses two complementary questions: how to rank expert sensitivity and how to rebalance a fixed average expert-bit budget across layers. Across Qwen1.5-MoE-2.7B and Qwen2-57B-A14B, Fisher-guided rebalancing improves an RTN quantization pipeline over uniform per-layer allocation and activation-frequency allocation under the same evaluation setup. The main takeaways are deliberately scoped: Fisher trace is a useful sensitivity signal in this setting, cross-layer rebalancing can recover substantial PPL under low-bit RTN, the preferred rebalancing strength differs across the two Qwen-MoE scales, and PPL alone may not predict downstream accuracy. We hope these results encourage further work on MoE-aware compression with stronger quantizers and deployment-ready mixed-precision kernels.

Limitations

This work has several important limitations, and we state them explicitly to clarify the intended scope of the claims.

1. **RTN-only quantization scope.** Our experiments are built around a per-channel RTN simulated-quantization pipeline. The paper should therefore be read as a study of *allocation under RTN*, not as evidence that RTN is preferable to stronger PTQ algorithms. Advanced quantizers such as GPTQ, AWQ, SmoothQuant, or QuIP# may produce better absolute accuracy when they are fully supported for the target MoE architecture.

2. **No GPTQ/AWQ comparison.** We do not report GPTQ or AWQ baselines because current public tooling does not yet provide stable support for expert-granular, cross-layer mixed-precision MoE execution. Reporting partial or unstable runs would be misleading. A unified comparison once MoE mixed-precision kernels mature is an important next step.

3. **Simulated quantization, not deployment.** We evaluate quantization error by dequantizing integer weights into bf16 tensors during forward propagation. This isolates the algorithmic effect of bit allocation but does not demonstrate wall-clock speedup or kernel-level deployability. Memory savings are theoretical estimates based on INT storage of expert weights.

4. **Model-family coverage.** Our empirical claims are limited to the two Qwen-MoE models studied. Other MoE designs, such as Mixtral-style smaller expert counts or DeepSeek-style fine-grained experts, may have different routing distributions, layer sensitivities, and optimal allocation patterns. We therefore avoid claiming a universal scaling law.

5. **Calibration and bitwidth design.** Fisher scores are estimated from WikiText-2 calibration batches, and bit choices are restricted to $\{2, 3, 4\}$. Domain-specific calibration data, alternative normalization choices, fractional bitwidths, or learned allocation policies may improve the rate–distortion trade-off.

References

Guanjie Chen, Xinyu Zhao, Tianlong Chen, and Yu Cheng. 2024. Moe-rbench: Towards building reliable language models with sparse mixture-of-experts. *arXiv:2406.11353*.

Mohammed Nowaz Rabbani Chowdhury, Kaoutar El Maghraoui, Hsinyu Tsai, Naigang Wang, Geoffrey W. Burr, Liu Liu, and Meng Wang. 2026. Efficient quantization of mixture-of-experts with theoretical generalization guarantees. In *The Fourteenth International Conference on Learning Representations*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. 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)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457*.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22. Curran Associates Inc.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23. Curran Associates Inc.

Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. 2020. Hawq-v2: hessian aware trace-weighted quantization of neural networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20. Curran Associates Inc.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. 2019. Hawq: Hessian aware quantization of neural networks with mixed-precision. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 293–302.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23(1).

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: massive language models can be accurately pruned

in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

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

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation. *Zenodo*.

Md Arafat Hossain, Xingfu Wu, Valerie Taylor, and Ali Jannesari. 2026. Generalizing scaling laws for dense and sparse large language models. *Preprint*, arXiv:2508.06617.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. Mixtral of experts. *arXiv:2401.04088*.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Guangxuan Xiao, and Song Han. 2025. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *GetMobile: Mobile Comp. and Comm.*, 28(4).

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. 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*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Guangshuo Qin, Zhiteng Li, Zheng Chen, Weihang Zhang, Linghe Kong, and Yulun Zhang. 2026. Veq: Modality-adaptive quantization for moe vision-language models. *Preprint*, arXiv:2602.01037.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9).

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. 2024. Quip#: even better llm quantization with hadamard incoherence and lattice codebooks. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

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. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, and 25 others. 2024. Qwen2.5 technical report. *arXiv:2412.15115*.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. Zeroquant: efficient and affordable post-training quantization for large-scale transformers. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22. Curran Associates Inc.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics*.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. Moefication: Transformer feed-forward layers are mixtures of experts. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 877–890, Dublin, Ireland. Association for Computational Linguistics.

Barret Zoph. 2022. Designing effective sparse expert models. In *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 1044–1044.