Profiling-Free Mixed-Precision Quantization for MoE LLMs via Fuzzy Rule Interpolation

Huachen Qi[1], **Ruiyu Zhuo**[1], **Bowen Shi**[1], **Xiang Chang**[1],
Fei Chao[1,2,*], **Changjing Shang**[2], **Qiang Shen**[2]
[1]School of Informatics, Xiamen University, Fujian, China
[2]Department of Computer Science, Aberystwyth University, Aberystwyth, UK
[*]Corresponding author: fchao@xmu.edu.cn

Abstract

Large Language Models continue to scale in size and capability, driving substantial computational and memory demands. Mixture-of-Experts (MoE) architectures alleviate this cost by activating only a sparse subset of experts per token, enabling efficient scaling without proportional increases in inference compute. However, quantization in MoE models remains challenging due to heterogeneous sensitivity across experts and their internal linear layers. Existing mixed-precision frameworks such as Mixed-precision Quantization for MoE (MxMoE) require full quantization-loss evaluation for expert–layer–and-bit configurations, incurring prohibitive profiling cost. To address this, we propose **FRI-MxMoE**, a profiling-free mixed-precision quantization framework that reformulates MoE calibration from exhaustive expert-wise profiling to sparse anchor profiling followed by Fuzzy Rule Interpolation. By constructing a fuzzy rule base in the intra-expert layer feature space (bit-width, activation variance, parameter scale), our method predicts quantization error from only sparse samples while remaining compatible with existing mixed-precision allocation objectives. Extensive experiments demonstrate that FRI-MxMoE accelerates the profiling phase by up to **15.7**× (on DeepSeek-V2) while achieving comparable or slightly superior zero-shot accuracy (e.g., **+1.04%** on DeepSeekV2-Lite) compared to the baseline. This enables continuous sensitivity modeling, preserves accuracy under mixed-precision allocation, and reduces offline computation by orders of magnitude. [1]

1 Introduction

The rapid progress of large language models (LLMs) has been driven by scaling both model size and training data. However, dense Transformer architectures incur rapidly growing computation and memory costs as they scale, making further expansion increasingly impractical. Mixture-of-Experts (MoE) architectures offer an effective alternative by activating only a small subset of experts per token, enabling conditional computation and substantially improved parameter efficiency (Lepikhin et al., 2020; Du et al., 2022; Lv et al., 2026). This sparse design allows MoE-based LLMs to maintain high model capacity at a fraction of the computational cost, and has therefore become a key building block in recent large-scale language models (Fedus et al., 2022; Rajbhandari et al., 2022).

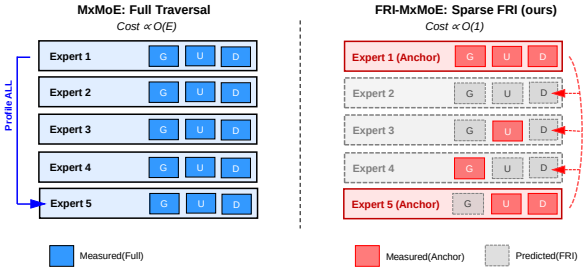

Figure 1: Comparison of Profiling Strategies.

Despite the structural efficiency of MoE architectures, achieving efficient and robust quantization remains a significant open challenge (Xie et al., 2025). Mixed-precision quantization—allocating different bitwidths across experts and layers—offers great potential for compression and acceleration, yet determining the optimal bit allocation is far from trivial (Tao et al., 2025). This difficulty arises from the inherently heterogeneous quantization sensitivity of MoE models, which manifests at multiple structural levels, while quantization can also induce non-trivial degradation and reliability issues (e.g., confidence/calibration) that complicate robust deployment (Zhao et al., 2024; Proskurina et al., 2024).

First, different experts respond unevenly to quantization noise because they learn distinct functions and exhibit highly variable activation frequen-

[1]Our code is available at https://github.com/qi-h-c/Fri-MxMoE.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 21484–21499

cies (Lo et al., 2025; Wang et al., 2025). Several experts are frequently routed to and dominate the model's overall behavior, whereas others remain rarely activated (Qiu et al., 2025; Huang et al., 2024a). Traditional mixed-precision quantization approaches, such as HAQ (Wang et al., 2019) and OMPQ (Ma et al., 2023a), largely assumes homogeneous quantization sensitivity across layers or model components. As a result, these methods struggle to adapt to the highly heterogeneous robustness exhibited by different experts and their internal linear sub-layers in MoE architectures (Xie et al., 2025; Chen et al., 2025).

Second, even within the same expert, different linear sub-layers—such as the gating projection, up-projection, and down-projection—exhibit distinct robustness to precision reduction (Xie et al., 2025; Cho et al., 2025). These sub-layers differ substantially in functional roles, parameter scales, and activation statistics, leading to highly heterogeneous quantization sensitivity within a single expert (Xie et al., 2025; Wei et al., 2023; Huang et al., 2024b; Ma et al., 2024b). However, many existing post-training and mixed-precision quantization approaches model quantization effects at a coarse granularity, treating linear layers uniformly without distinguishing their roles within an expert (Xie et al., 2025; Liao et al., 2024). Methods such as AdaRound (Nagel et al., 2020), while effective for dense models, do not explicitly capture this intra-expert, sub-layer-level heterogeneity, which becomes particularly pronounced in large MoE architectures (Xie et al., 2025; Chen et al., 2025; Ma et al., 2023b).

Third, the highly uneven activation frequency of experts introduces an additional challenge for precision allocation in MoE models (Qiu et al., 2025; Huang et al., 2024a). Due to sparse routing, only a small subset of experts is activated for each token, and the resulting expert usage distribution is often heavily skewed (Lo et al., 2025; Wang et al., 2025). Consequently, quantization errors in frequently activated experts have a much larger impact on overall model quality than those in rarely used ones (Chen et al., 2025; Xie et al., 2025). However, many profiling-based mixed-precision methods implicitly assign equal importance to all experts when estimating quantization loss, regardless of their activation frequency (Xie et al., 2025). This uniform treatment leads to inefficient resource allocation, as substantial profiling effort is wasted on rarely activated experts that contribute little to end-to-end performance (Qiu et al., 2025; Huang et al., 2024a). Similar observations have been reported in recent MoE analyses, which highlight the importance of expert activation patterns for both optimization efficiency and model quality (Fedus et al., 2022; Du et al., 2022; Lo et al., 2025; Qiu et al., 2025).

To address the first issue, FRI-MxMoE employs a profiling-free strategy via Fuzzy Rule Interpolation (FRI). Instead of exhaustively profiling all experts, we measure only a few 'anchor' experts and interpolate the sensitivity of the others (Figure 1), significantly reducing calibration effort. This is more than replacing a predictor inside the same pipeline: MxMoE builds a dense expert-wise sensitivity table by traversal, whereas FRI-MxMoE builds a sparse anchor-derived rule base and queries a continuous sensitivity manifold on demand. The profiling stage itself is therefore reformulated from exhaustive enumeration to sparse profiling plus interpolation.

For the second issue, FRI-MxMoE constructs a fuzzy rule base within the intra-expert feature space. By incorporating bit-width, activation variance, and parameter scale into rule antecedents, FRI continuously models non-linear quantization sensitivity across sub-layers, offering flexible and interpretable precision predictions.

Third, FRI-MxMoE integrates a frequency-aware weighting mechanism directly into the fuzzy inference process and employs Lagrangian relaxation for efficient optimization. This ensures that quantization errors in frequently activated experts are prioritized while yielding fast, scalable allocation solutions.

In summary, our contributions are threefold:

(1) FRI-based Heterogeneous Sensitivity Modeling. We propose a FRI mechanism to model expert sensitivity from sparse samples. By learning the mapping from expert features to quantization error, our method enables accurate, profiling-free estimation, avoiding exhaustive profiling.

(2) Fine-grained Intra-Expert Modeling. We extend FRI to capture the distinct robustness of intra-expert sub-layers (gate, up, down). This allows for fine-grained, structure-aware mixed-precision allocation that respects internal expert architecture.

(3) Efficient Frequency-aware Allocation. We introduce a frequency-aware objective solved via Lagrangian relaxation on the smooth FRI-predicted landscape. This approach offers superior scalability and noise robustness compared to ILP solvers,

yielding fast, high-quality allocations.

To our knowledge, this is the first attempt to apply FRI to the field of neural network quantization, specifically to address the scalability challenges in profiling large-scale MoE models.

2 Related Work

Fuzzy Rule Interpolation (FRI) was introduced to support inference with sparse or incomplete rule bases (Kóczy and Hirota, 1993b,a; Kovács and Kóczy, 1997). Rather than requiring dense rule coverage as in Mamdani or Takagi–Sugeno systems, FRI interpolates among the most relevant neighbouring rules to produce outputs even when no rule exactly matches the input. Recent advances improve robustness and stability in high-dimensional spaces, including density-aware neighbour selection (Gao, 2023) and general fuzzy-set representations for higher-quality interpolation (Qu et al., 2024). FRI has also been extended to dynamic systems, predictive control, and industrial signal recovery, showing strong capability in approximating nonlinear functions under limited knowledge (Baranyi et al., 2004; Jiang et al., 2025).

Post-Training Quantization (PTQ) is widely used to compress large language models without retraining. Methods such as AdaRound (Nagel et al., 2020), GPTQ (Frantar et al., 2023), and AffineQuant (Ma et al., 2024a) leverage reconstruction or equivalent transformations to reduce layer-wise quantization error, while activation-aware approaches like AWQ (Lin et al., 2024) preserve salient weights identified by activation statistics. Recent analyses also study oscillation behavior in PTQ and outlier-aware reconstruction for Transformer-style models (Ma et al., 2023b, 2024b). Although effective for dense backbones (e.g., LLaMA/OPT), extending PTQ to MoE is non-trivial due to the enlarged parameter space and expert-specific behaviors.

MoE models (e.g., Mixtral, Switch Transformer) employ sparse routing where only a few experts are activated per token. Recent work has begun targeting MoE quantization: QMoE (Frantar and Alistarh, 2023) enables low-bit quantization of massive MoE models using scalable calibration and efficient GPU kernels. However, most existing strategies still use uniform precision or treat experts equivalently, ignoring *heterogeneous quantization sensitivity* driven by expert specialization and skewed activation frequency. While mixed-precision could exploit this heterogeneity, identifying an optimal per-expert bit-width typically requires profiling every expert, incurring a prohibitive cost that scales as $O(E)$ with the number of experts.

3 Method

3.1 Mixed-precision Quantization for MoE

MxMoE is a state-of-the-art mixed-precision quantization framework designed for MoE models. It formulates the bit-width allocation problem as a constrained optimization task, aiming to maximize model performance (e.g., minimize perplexity degradation) under a strict latency or memory budget. Formally, let an MoE model have L layers and E experts per layer. Each expert consists of linear sub-layers (e.g., gate, up, down projections). MxMoE assigns a bit-width $b_{i,j,k} \in \mathcal{B}$ to the k-th sub-layer of the j-th expert in the i-th layer, where $\mathcal{B}$ is a candidate set (e.g., $\{2, 4, 8\}$ bits).

To solve this optimization problem, MxMoE requires a error table $\mathcal{T}$, which records the quantization error $\mathcal{L}(b_{i,j,k})$ for every possible configuration. Typically, this is obtained by profiling: quantizing each block individually and measuring the reconstruction loss (e.g., MSE) on a calibration dataset.

Profiling Bottleneck: While effective, MxMoE faces a severe scalability challenge. The cost of constructing the full sensitivity table $\mathcal{T}$ scales linearly with the number of experts:

$$C_{\text{profile}} \propto L \times E \times |\mathcal{B}| \times N_{\text{calib}}, \tag{1}$$

where N_{calib} is the number of calibration samples. For modern MoE models such as Mixtral-8x7B ($E = 8$) or Switch Transformer (E up to hundreds), C_{profile} becomes prohibitively expensive. Furthermore, if we consider fine-grained runtime constraints (e.g., varying hardware tile sizes M, N, K), the search space explodes, making brute-force profiling impossible.

We observe that the quantization sensitivity of experts is not random. It exhibits strong correlations with expert-specific features (e.g., weight scale, activation frequency) and structural parameters (e.g., layer type, bit-width). This motivates us to replace the dense profiling with an analytical prediction model.

3.2 FRI-MxMoE: Fuzzy Rule Interpolation for Scalable Quantization

Figure 2 illustrates the proposed FRI-MxMoE. Starting from a pre-trained MoE model and cal-

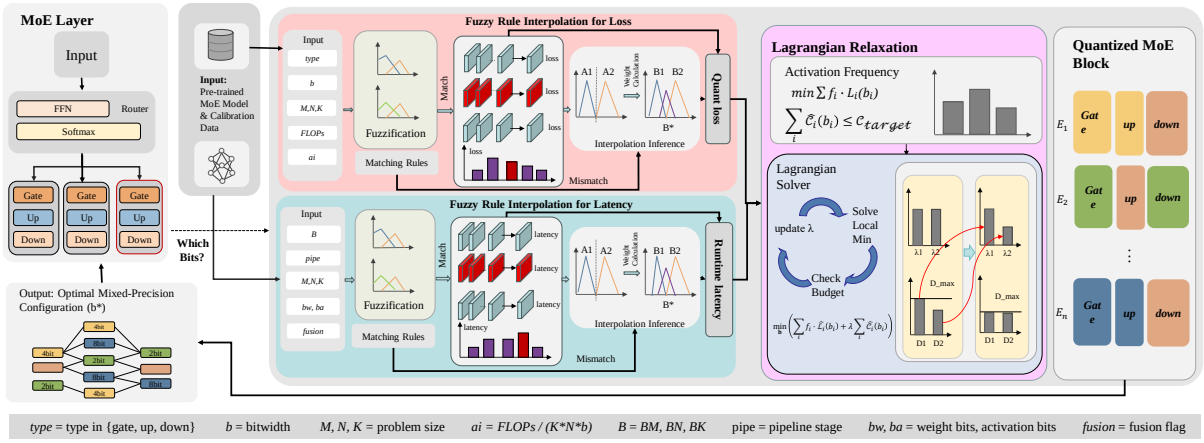

Figure 2: Overview of proposed FRI-MxMoE framework. Quantization error and latency are predicted via fuzzy rule interpolation and used for mixed-precision optimization.

ibration data, we profile only sparse anchor configurations and convert each profiled expert/sub-layer/bit tuple into a lightweight descriptor ($type$, (M, N, K), $FLOPs$, and $(w_{\text{bits}}, a_{\text{bits}})$). Two parallel FRI modules then predict loss and latency for unprofiled configurations across candidate bit-widths.

These predicted surfaces are optimized via Lagrangian relaxation with activation-frequency weighting, iteratively updating λ and solving local subproblems to satisfy the target budget. The output is a per-expert and per-sub-layer bit-width assignment b^* with balanced quality and efficiency.

3.2.1 Fuzzification of Quantization Variables

We define distinct sets of linguistic variables to capture the specific characteristics of quantization error and runtime latency. For a detailed definition of these variables, please refer to Table 7 in Appendix A.

Briefly, for **Quantization Error Interpolation**, the input vector $\mathbf{x}_{err}$ includes structural type, bit-width, scale features, and arithmetic intensity proxy. For **Runtime Latency Interpolation**, the input vector $\mathbf{x}_{lat}$ incorporates hardware-aware kernel parameters such as tiling configuration, pipeline stages, and fusion status.

For continuous variables (e.g., bit-width, M, N, K), we employ triangular membership functions defined by a set of breakpoints $\{a_k\}$. The membership degree $\mu_{A_k}(x)$ of value x to the linguistic term A_k is:

$$\mu_{A_k}(x) = \max\left(0, 1 - \frac{|x - c_k|}{w_k}\right), \qquad (2)$$

where c_k is the center and w_k is the width of the triangle. For categorical variables (e.g., weight type), we use singleton (one-hot) membership functions.

3.2.2 Sparse Rule Base Construction

We collect a small set of anchor samples $\mathcal{S} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N_s}$. To capture the non-uniform sensitivity distribution across model depths, we employ a **Depth-Adaptive Stratified Sampling** strategy. We divide the model layers into three stages (shallow, middle, deep) and apply varying sampling densities based on their sensitivity contribution. For instance, we sample experts more densely in the middle layers (e.g., selecting every second expert) compared to the shallow (e.g., selecting every fourth expert) and deep layers.

Crucially, our profiling penetrates into the *expert internal sub-layers*: for each selected expert, we individually profile its Gate, Up, and Down projections under a set of key bit-widths (e.g., $\{2, 3, 4, 8\}$). Each sample is converted into a fuzzy rule R_i:

$$R_i : \textbf{IF } x_1 \text{ is } A_1^{(i)} \wedge \cdots \wedge x_d \text{ is } A_d^{(i)} \textbf{ THEN } y \text{ is } y^{(i)}. \qquad (3)$$

If N_a anchor experts are selected and each contributes $C{=}3$ sub-layers and $|\mathcal{B}_p|$ profiled bit-widths, then the rule-base size is $N_s = N_a \times C \times |\mathcal{B}_p|$. In our setup, $\mathcal{B}_p = \{2, 3, 4, 8\}$, so a 5k–10k rule base corresponds to profiling only about 417–833 anchor experts in total across the whole model. This is still far smaller than exhaustive traversal, and Table 3 later shows that prediction quality already saturates around 7.5k–10k rules. In practice, we start from a 5k budget and increase it only when held-out rank correlation keeps improving. Unlike traditional fuzzy systems that require a dense grid

of rules to cover the entire input space, our rule base is "sparse". Most input configurations will not strictly "match" any existing rule (i.e., activation strength is zero).

3.2.3 Fuzzy Interpolation Mechanism

To handle the sparsity, we employ an FRI inference engine based on Inverse Distance Weighting (IDW) in the feature space. Given a query configuration $\mathbf{x}_q$, we first calculate its distance to each rule antecedent R_i. We define a heterogeneous distance metric $D(\mathbf{x}_q, R_i)$ that combines Euclidean distance for continuous features and Hamming distance for categorical features:

$$\begin{aligned} D(\mathbf{x}_q, R_i) &= \sqrt{D_{\text{cont}}(\mathbf{x}_q, R_i) + D_{\text{cat}}(\mathbf{x}_q, R_i)}, \\ D_{\text{cont}}(\mathbf{x}_q, R_i) &= \sum_{j \in \text{cont}} \left(\frac{x_{q,j} - c_{i,j}}{\sigma_j} \right)^2, \\ D_{\text{cat}}(\mathbf{x}_q, R_i) &= \sum_{j \in \text{cat}} \mathbb{I}(x_{q,j} \neq c_{i,j}). \end{aligned} \tag{4}$$

The predicted quantization error (or runtime) $\hat{y}$ is computed by interpolating the K nearest rules ($N_K(\mathbf{x}_q)$):

$$\hat{y} = \frac{\sum_{R_i \in N_K(\mathbf{x}_q)} w_i \cdot y^{(i)}}{\sum_{R_i \in N_K(\mathbf{x}_q)} w_i}, \quad w_i = \frac{1}{D(\mathbf{x}_q, R_i) + \epsilon}. \tag{5}$$

where ϵ is a small constant to prevent division by zero.

Stability of IDW-FRI. Let $\alpha_i = w_i / \sum_{R_j \in N_K(\mathbf{x}_q)} w_j$. Then $\hat{y} = \sum_i \alpha_i y^{(i)}$ with $\alpha_i \geq 0$ and $\sum_i \alpha_i = 1$, so the prediction is bounded by its neighbours: $\min_i y^{(i)} \leq \hat{y} \leq \max_i y^{(i)}$. If anchor outputs are perturbed by bounded noise $y^{(i)} + \delta_i$, the induced prediction error satisfies $|\delta\hat{y}| = |\sum_i \alpha_i \delta_i| \leq \max_i |\delta_i|$. Moreover, with $\epsilon > 0$ and normalized inputs, $D(\mathbf{x}_q, R_i)$ and w_i are smooth away from exact collisions, which makes $\hat{y}$ locally Lipschitz in $\mathbf{x}_q$. These properties do not constitute a global end-task guarantee, but they explain why IDW-FRI yields a smooth sensitivity surface and is resistant to isolated noisy anchors. A proof sketch is provided in Appendix F. This mechanism allows FRI-MxMoE to smoothly generalize the sensitivity patterns learned from a few experts to the entire MoE model, enabling accurate error prediction for unprofiled configurations.

3.2.4 Compute-Aware Enhancement

A unique feature of our implementation is the injection of compute-related priors (M, N, K, FLOPs) into the interpolation. Standard bit-width interpolation often fails to capture the hardware efficiency cliffs (e.g., the performance gap between fitting in L2 cache vs. DRAM).

By explicitly including these physical dimensions in the fuzzy distance metric $D(\cdot)$, FRI-MxMoE can distinguish between compute-bound and memory-bound layers, yielding more robust runtime and error estimates for diverse hardware backends.

3.2.5 Lagrangian-based Resource Allocation

After estimating the quantization error $\hat{\mathcal{L}}$ and cost $\hat{\mathcal{C}}$ via FRI for the full configuration space, we aim to find the optimal bit-width assignment $\mathbf{b}^*$. A critical observation in MoE models is the highly skewed expert activation distribution—a small subset of experts handles the majority of tokens. Treating all experts equally would lead to suboptimal resource utilization.

To address this, we introduce a **Frequency-Aware Weighting** mechanism into the optimization objective. Let f_i denote the normalized activation frequency of the i-th quantization unit (obtained from calibration data). The weighted optimization problem is defined as:

$$\min_{\mathbf{b}} \sum_i f_i \cdot \hat{\mathcal{L}}_i(b_i) \quad \text{s.t.} \quad \sum_i \hat{\mathcal{C}}_i(b_i) \leq \mathcal{C}_{\text{target}}, \tag{6}$$

where i indexes each quantization unit and $b_i \in \mathcal{B}$. The term $f_i \cdot \hat{\mathcal{L}}_i(b_i)$ ensures that frequently activated experts are prioritized for higher precision, while rarely used experts can be compressed more aggressively without degrading overall performance.

To ensure scalability, we relax the hard constraint into the objective function using a Lagrange multiplier $\lambda \geq 0$:

$$\min_{\mathbf{b}} \left(\sum_i f_i \cdot \hat{\mathcal{L}}_i(b_i) + \lambda \sum_i \hat{\mathcal{C}}_i(b_i) \right). \tag{7}$$

For a fixed λ, the global optimization decouples into independent local selections for each unit i:

$$b_i^*(\lambda) = \arg\min_{b \in \mathcal{B}} \left(f_i \cdot \hat{\mathcal{L}}_i(b) + \lambda \hat{\mathcal{C}}_i(b) \right). \tag{8}$$

Since the total cost $\sum \hat{\mathcal{C}}_i(b_i^*(\lambda))$ is monotonically non-increasing with respect to λ, we efficiently find

the optimal λ^* satisfying the budget constraint via binary search.

To avoid confusion between the number of experts and the number of decision variables, we denote by U the total number of allocation units (e.g., expert sub-layers) whose bit-widths are to be selected, where typically $U = L \times E \times C$. Our Lagrangian relaxation solves the budget-constrained allocation via a binary search on λ. Each iteration requires a single pass over all U units to calculate the induced cost and update λ. Therefore, the solver runs in $O(N_{\text{iter}} \cdot U)$ time, where N_{iter} is the number of binary-search iterations (a small constant in practice). Importantly, this allocation step is lightweight compared to quantized forward-pass profiling.

Note that our "$O(1)$ profiling" statement refers to the *calibration/profiling* cost being independent of the number of experts E, whereas the subsequent mixed-precision allocation is solved via a Lagrangian binary search with runtime $O(N_{\text{iter}} \cdot U)$ in the number of allocation units U.

We summarize the overall pipeline of FRI-MxMoE in Algorithm 1, which is provided in Appendix B.

4 Experimentation

We conduct our experiments on a server equipped with NVIDIA A800 GPUs. We focus on the calibration bottleneck of mixed-precision MoE quantization. The error-FRI is hardware-agnostic, while the latency-FRI is hardware-conditional because it explicitly consumes device/kernel descriptors; porting to a new backend therefore requires only sparse anchor profiling on that backend rather than exhaustive traversal. We use the Wikitext-2 (Merity et al., 2016) dataset for calibration, randomly sampling 128 sequences with a sequence length of 4096. For zero-shot evaluation, we employ seven standard benchmarks: Arc-Challenge (AC) (Clark et al., 2018), Arc-Easy (AE) (Clark et al., 2018), HellaSwag (HS) (Zellers et al., 2019), LAMBADA-openai (LO) (Paperno et al., 2016), LAMBADA-standard (LS) (Paperno et al., 2016), PIQA (PQ) (Bisk et al., 2020), and WinoGrande (WG) (Sakaguchi et al., 2021), alongside perplexity (PPL) measurement on Wikitext-2.

In Table 1, we report the accuracy for each task and the average accuracy (Avg.) across all seven tasks. In Table 2, we note that the reported profiling time and cost correspond to a single quantization configuration scan (e.g., W4A4). In a practical mixed-precision search scenario, multiple such scans (for different candidate bit-widths) would be required, further amplifying the efficiency advantage of our profiling-free approach.

4.1 Performance Analysis

Table 1 presents the zero-shot performance on seven standard benchmarks and perplexity (PPL) on Wikitext-2. The baseline MxMoE employs an Integer Linear Programming (ILP) solver for precision allocation, whereas our FRI-MxMoE utilizes a fuzzy-logic-based allocation strategy.

Table 1 indicates that FRI-MxMoE consistently outperforms existing methods across diverse model architectures. Compared to the standard MxMoE, our approach achieves slightly better accuracy (e.g., **+0.04%** on DeepSeekV2-Lite and **+0.03%** on Mixtral-8×7B under W3.25-A16) and lower perplexity. This marginal yet consistent superiority stems from the manifold continuity inherent in FRI.

Unlike MxMoE, which optimizes over a discrete and potentially noisy set of profiled points, FRI constructs a smooth error surface that captures the underlying sensitivity trends. Coupled with the global search capability of our Lagrangian solver, FRI-MxMoE avoids local optima often encountered by greedy heuristics in discrete spaces, effectively "denoising" the sensitivity map to find more robust allocation strategies. Overall, these results show that profiling efficiency is gained without sacrificing downstream task quality.

4.2 Efficiency Analysis

We compare the profiling cost of FRI-MxMoE against the baseline MxMoE across different model scales in Table 2. MxMoE requires exhaustively profiling all expert-layer-strategy combinations to construct the sensitivity table. The search space size is defined as $L \times E \times 3 \times |\mathcal{S}|$, where L is layers, E is experts, and $|\mathcal{S}|$ is the number of candidate strategies.

Comparison between Scalability and Model Characteristics: The results highlight a fundamental shift in complexity class. MxMoE's cost scales linearly with the number of experts ($O(E)$), leading to prohibitive times for fine-grained MoEs like DeepSeek-V2 (estimated >190 hours). In contrast, once the anchor budget is fixed, FRI-MxMoE has approximately constant profiling cost with respect to E. This is because FRI-MxMoE compresses

Table 1: Zero-shot performance on standard benchmarks. We report accuracy (%) for Arc-Challenge (AC), Arc-Easy (AE), HellaSwag (HS), LAMBADA-openai (LO), LAMBADA-standard (LS), PIQA (PQ), and WinoGrande (WG). PPL denotes perplexity on Wikitext-2. "Avg." is the average accuracy across the seven tasks. GPTQ* denotes GPTQ with random Hadamard transformation. FRI-MxMoE (Ours) achieves comparable accuracy to MxMoE with significantly reduced allocation cost.

Model	Method	#Bits (W-A)	AC	AE	HS	LO	LS	PQ	WG	Avg.	PPL
DeepSeekV2-Lite	Baseline	16-16	48.98	76.22	77.91	72.33	67.90	80.20	71.19	70.68	5.92
	GPTQ*	3.25-16	47.35	75.04	76.44	70.41	65.65	79.05	71.27	69.32	6.18
	GPTQ*	2.25-16	37.63	63.47	65.45	52.53	48.55	74.59	64.09	58.04	8.49
	QuaRot	4-4	41.81	67.51	74.12	50.01	45.86	75.52	63.38	59.74	8.44
	MxMoE	3.25-16	47.87	74.58	76.85	71.10	65.85	79.27	70.09	69.37	6.08
	MxMoE	2.25-16	40.36	68.86	68.63	59.56	54.01	75.08	67.80	62.04	**7.01**
	MxMoE	5-5	46.76	74.37	77.38	68.41	64.99	79.38	69.22	68.64	6.16
	FRI-MxMoE	3.25-16	48.91	75.61	77.87	72.16	66.89	80.28	71.12	**70.41**	**5.96**
	FRI-MxMoE	2.25-16	40.89	69.38	69.17	60.09	54.57	75.62	68.53	**62.58**	7.09
	FRI-MxMoE	5-5	46.78	74.41	77.41	68.47	65.01	79.41	69.26	**68.68**	**6.11**
Qwen1.5-MoE	Baseline	16-16	44.03	69.53	77.26	71.28	64.62	80.47	69.30	68.07	6.79
	GPTQ*	3.25-16	43.34	68.60	75.35	68.68	62.80	79.22	66.54	66.36	7.15
	GPTQ*	2.25-16	30.89	47.14	60.77	43.72	34.81	69.97	56.20	49.07	11.19
	QuaRot	4-4	27.13	40.74	57.10	35.61	25.33	66.43	51.93	43.47	18.44
	MxMoE	3.25-16	43.77	66.04	75.92	69.71	62.82	79.11	68.03	66.49	7.02
	MxMoE	2.25-16	31.66	53.28	62.80	56.43	51.00	71.33	61.25	55.39	8.79
	MxMoE	5-5	42.92	66.04	76.27	70.06	63.40	80.58	67.80	66.72	7.01
	FRI-MxMoE	3.25-16	43.93	66.17	76.06	69.83	62.99	79.23	68.17	**66.63**	**6.94**
	FRI-MxMoE	2.25-16	31.68	53.34	62.83	56.47	51.02	71.38	61.28	**55.43**	**8.67**
	FRI-MxMoE	5-5	43.16	66.26	76.53	70.29	63.62	80.82	68.03	**66.96**	**6.92**
Qwen2-MoE	Baseline	16-16	55.20	77.19	84.09	74.35	62.62	82.32	72.14	72.56	5.84
	GPTQ*	3.25-16	53.67	75.88	82.90	73.36	63.24	81.01	70.96	**71.57**	**6.11**
	GPTQ*	2.25-16	38.82	57.66	71.27	58.99	49.72	73.29	60.30	58.58	7.98
	QuaRot	4-4	33.19	42.72	54.34	23.02	9.53	63.87	50.12	39.54	110.66
	MxMoE	3.25-16	53.84	76.30	82.81	72.39	60.95	81.34	69.69	71.05	6.18
	MxMoE	2.25-16	45.05	68.86	77.13	66.00	56.61	75.41	62.90	64.57	7.57
	MxMoE	5-5	54.86	75.55	82.69	72.87	62.68	79.49	70.96	71.30	6.25
	FRI-MxMoE	3.25-16	53.93	76.40	82.89	72.50	61.04	81.42	69.79	71.14	6.23
	FRI-MxMoE	2.25-16	45.18	68.98	77.27	66.13	56.76	75.53	63.04	**64.70**	**7.44**
	FRI-MxMoE	5-5	54.88	75.58	82.73	72.89	62.71	79.51	70.99	**71.33**	**6.20**
Mixtral-8×7B	Baseline	16-16	66.38	85.39	85.95	77.28	73.06	85.20	76.72	78.57	3.88
	GPTQ*	3.25-16	64.42	84.01	85.12	76.77	71.76	83.79	76.16	77.43	4.17
	GPTQ*	2.25-16	48.89	72.35	76.95	68.39	61.44	77.15	67.72	67.56	5.69
	QuaRot	4-4	50.60	68.69	75.65	40.95	38.83	76.88	61.01	58.94	9.06
	MxMoE	3.25-16	64.25	84.22	85.04	76.98	71.86	84.17	75.93	77.49	4.15
	MxMoE	2.25-16	48.98	72.77	77.44	68.68	62.18	76.28	68.90	67.89	5.63
	MxMoE	5-5	64.08	83.71	85.10	76.21	71.78	83.79	73.80	76.92	**4.20**
	FRI-MxMoE	3.25-16	64.31	84.27	85.10	77.05	71.92	84.22	76.00	**77.55**	**4.11**
	FRI-MxMoE	2.25-16	49.06	72.83	77.51	68.76	62.25	76.37	68.96	**67.96**	**5.61**
	FRI-MxMoE	5-5	65.11	84.75	86.12	77.24	72.82	84.82	74.83	**77.96**	4.22

Table 2: Architectural Specifications and Profiling Efficiency. We compare the profiling cost of FRI-MxMoE against MxMoE on extended MoE variants. "Params" denotes model size in GB. "Experts" indicates Sparse+Shared experts. FRI-MxMoE demonstrates superior scalability on fine-grained MoEs (Qwen, DeepSeek) and massive models (DeepSeek-V2), despite a slight overhead on dense-expert models like Mixtral-8×22B.

Model Variant	Params (GB)	Experts	TopK	Search Space	MxMoE Time	FRI Time	Speedup
Mixtral-8×7B-Instruct-v0.1	92.9	8	2	6,912	1h 41m	1h 01m	1.7×
Mixtral-8×22B-Instruct-v0.1	281.0	8	2	6,912	5h 07m	3h 02m	1.7×
Qwen1.5-MoE	26.7	60+4	4	38,880	2h 45m	42m	3.9×
Qwen2-MoE-Instruct	106.9	64+8	8	41,472	11h 45m	2h 48m	4.2×
DeepSeek-V2-Lite	29.3	64+2	6	46,656	3h 37m	46m	4.7×
DeepSeek-V2	472.0	160+2	6	155,520	194h 30m	12h 22m	**15.7**×

Table 3: Impact of rule-base size on FRI prediction quality on Qwen1.5-MoE. We evaluate ΔPPL prediction on a held-out profiled set.

Rule Base Size	MAE ↓	RMSE ↓	Spearman ρ ↑
1k	0.182	0.241	0.61
2.5k	0.131	0.178	0.69
5k	0.096	0.132	0.75
7.5k	0.082	0.114	0.78
10k	**0.076**	**0.106**	**0.80**

the high-dimensional sensitivity space into a low-dimensional manifold governed by physical priors (e.g., parameter count, activation frequency). By learning this mapping from a fixed number of anchor samples, we decouple profiling cost from model width. While this incurs a fixed overhead for parameter-heavy, expert-sparse models (e.g., Mixtral-8×22B, 1.7× speedup), the advantage grows quickly with expert count and reaches **15.7**× on DeepSeek-V2, making FRI-MxMoE particularly attractive for next-generation expert-rich MoEs.

4.3 Anchor Budget and Practicality

Table 3 shows clear diminishing returns as the rule-base size increases. A 5k rule base already preserves the sensitivity ranking well (ρ=0.75) with low prediction error, while 7.5k–10k is close to saturation. This makes rule-budget selection practical for a new model: start from a fixed 5k budget, profile a small held-out validation split, and expand only if the rank correlation has not stabilized. Importantly, the apparently large 4k–10k rule count does not imply profiling thousands of distinct experts, because each anchor expert contributes multiple rules through three sub-layers and multiple profiled bit-widths. Appendix E further shows that our depth-adaptive stratified sampling is more sample-efficient than random sampling under the same rule budget.

4.4 Robustness of Sensitivity Modeling

We also compare FRI against standard learned regressors under the same sparse-profiling budget. Table 4 shows that FRI achieves the best MAE/RMSE and the highest rank correlation, even though it requires no training. This supports our design choice: the anchor set is intentionally optimized for local interpolation rather than dense i.i.d. supervision, so a lightweight interpolation model is more sample-efficient than global regressors in the anchor-only regime targeted by our method.

Table 4: Comparison with learned regression predictors on Qwen1.5-MoE. All methods use the same anchor-only supervision and the same input features, and are evaluated on held-out profiled configurations using ΔPPL prediction.

Method	MAE ↓	RMSE ↓	Spearman ρ ↑
FRI (ours)	**0.096**	**0.132**	**0.75**
Ridge	0.118	0.161	0.69
SVR (RBF)	0.104	0.158	0.71
RF / GBDT	0.101	0.149	0.73
MLP	0.112	0.172	0.68

Table 5: Comparison of different allocation granularities. Test with 5-bits weight-activation quantization. Evaluation metrics are the same as described in settings.

Model	PPL ↓		Avg. Acc ↑	
	Linear	Expert	Linear	Expert
DeepSeek-V2-Lite	**6.11**	6.32	**68.68**	67.88
Qwen1.5-MoE	**6.92**	6.98	**66.96**	66.11

We further test whether FRI-MxMoE is brittle to the exact feature design. The detailed results in Table 10 and Table 11 show that removing activation statistics or weight-scale features causes only mild degradation across four MoE families, and replacing variance with semantically similar alternatives such as mean-absolute activation or outlier ratio remains close to the default setting. Together, these results indicate that FRI-MxMoE relies on semantically meaningful sensitivity cues rather than a fragile, model-specific feature choice.

4.5 Further Analysis

Effect of bitwidth allocation granularity. MxMoE employs linear-block level allocation instead of the expert-level allocation often found in previous studies. We also perform bitwidth allocation at the expert level for comparison, as shown in Table 5. The results demonstrate that linear-block allocation consistently outperforms expert-level allocation.

To validate the effectiveness of our proposed FRI and Lagrangian Relaxation strategies, we conduct an ablation study on Qwen1.5-MoE in Table 6.

Component Analysis: The ablation results reveal a critical synergy between FRI and Lagrangian relaxation. Using FRI alone with ILP yields competitive performance (**66.41** Avg), but replacing ILP with our Lagrangian solver further boosts accuracy to **66.63**. At first glance, this is counter-intuitive: why would an approximate solver (La-

Table 6: Ablation study on Qwen1.5-MoE. We compare four combinations of profiling methods (Full Traversal vs. FRI Interpolation) and allocation solvers (ILP vs. Lagrangian Relaxation). The combinations are: (1) MxMoE (Full + ILP), (2) Full + Lagrangian, (3) FRI + ILP, and (4) FRI-MxMoE (FRI + Lagrangian). Our proposed FRI-MxMoE achieves the best performance across all metrics.

Profiling	Solver	#Bits	AC	AE	HS	LO	LS	PQ	WG	Avg.	PPL
Full Traversal	ILP (MxMoE)	3.25-16	43.77	66.04	75.92	69.71	62.82	79.11	68.03	66.49	7.02
Full Traversal	Lagrangian	3.25-16	43.62	65.88	75.78	69.58	62.68	78.98	67.88	66.34	7.10
FRI Interpolation	ILP	3.25-16	43.70	65.95	75.85	69.65	62.75	79.05	67.95	66.41	7.06
FRI Interpolation	Lagrangian (Ours)	3.25-16	43.93	66.17	76.06	69.83	62.99	79.23	68.17	**66.63**	**6.94**
Full Traversal	ILP (MxMoE)	2.25-16	31.66	53.28	62.80	56.43	51.00	71.33	61.25	55.39	8.79
Full Traversal	Lagrangian	2.25-16	31.48	53.10	62.62	56.25	50.82	71.15	61.05	55.21	8.92
FRI Interpolation	ILP	2.25-16	31.58	53.20	62.72	56.35	50.92	71.25	61.15	55.31	8.85
FRI Interpolation	Lagrangian (Ours)	2.25-16	31.68	53.34	62.83	56.47	51.02	71.38	61.28	**55.43**	**8.67**

grangian) outperform an exact solver (ILP)? The answer lies in the *noise robustness* required when optimizing over a predicted surface: FRI predictions inevitably contain approximation errors, and ILP tends to exploit these local inaccuracies too aggressively. In contrast, Lagrangian relaxation searches for a global slope λ that balances the marginal benefit across all experts. This implicit consistency constraint acts as a form of *regularization*, preventing the selection of outlier configurations driven by local prediction noise. Thus, while ILP finds the optimal solution for the *predicted* landscape, Lagrangian relaxation yields a solution that is more robust and generalizable to the *true* performance landscape. FRI-MxMoE (Row 4) effectively combines the scalability of FRI with the robustness of Lagrangian relaxation.

5 Conclusion

In this work, we presented FRI-MxMoE, a scalable mixed-precision quantization framework for MoE-based LLMs that replaced exhaustive expert-wise profiling with FRI. By constructing a sparse fuzzy rule base over intra-expert layer features and interpolating quantization error (and cost) for unprofiled configurations, the proposed approach substantially reduced the calibration burden and enabled smooth, continuous sensitivity modeling under heterogeneous expert behaviors. We further incorporated frequency-aware weighting to prioritize high-usage experts during allocation, and adopted an FRI-compatible Lagrangian relaxation to efficiently solve the budget-constrained precision assignment at scale, avoiding the poor scalability of ILP-based solvers. Empirically, FRI-MxMoE consistently matched or slightly improved the accuracy–perplexity trade-off of strong MoE quantization baselines while delivering large reductions in profiling time, with particularly pronounced gains on fine-grained and massive MoE variants.

Limitations

Our approach alleviates the main scalability bottleneck of calibration/profiling by learning a sensitivity manifold from a fixed set of anchor samples, but it still has three limitations. (1) FRI-MxMoE introduces an interpolation overhead that is largely independent of the number of experts, so its end-to-end speedup is less pronounced when baseline profiling is already moderate, such as in parameter-heavy yet expert-sparse models like Mixtral-8$\times$22B, where the speedup is **1.7**$\times$. (2) Its interpolation accuracy depends on the quality of the anchor set and feature space; unrepresentative anchors or missing factors may lead to assignment errors, suggesting the need for adaptive anchor selection and richer feature design in more challenging settings. (3) The latency model is hardware-conditional rather than hardware-universal. Although the feature space can incorporate device-specific kernel descriptors, our results are reported only on A800 GPUs, and deployment on new hardware still requires sparse anchor profiling. In addition, the allocation stage is not constant-time, but scales as $O(N_{\text{iter}} \cdot U)$ due to binary search over the Lagrange multiplier, although it remains much cheaper than full profiling.

Acknowledgments

This work was supported by the Key Program of the National Natural Science Foundation of China Joint Fund (No. U23A20383) and Xiamen Municipal Natural Science Foundation Project (No. 3502Z202573010).

References

P. Baranyi, L.T. Koczy, and T.D. Gedeon. 2004. A generalized concept for fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, 12(6):820–837.

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439.

Yuanteng Chen, Yuantian Shao, Peisong Wang, and Jian Cheng. 2025. EAC-MoE: Expert-selection aware compressor for mixture-of-experts large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12942–12963, Vienna, Austria. Association for Computational Linguistics.

Yoonjun Cho, Soeun Kim, Dongjae Jeon, Kyelim Lee, Beomsoo Lee, and Albert No. 2025. Assigning distinct roles to quantized and low-rank matrices toward optimal weight decomposition. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14453–14470, Vienna, Austria. 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. *Preprint*, arXiv:1803.05457.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, and 8 others. 2022. GLaM: Efficient scaling of language models with mixture-of-experts. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5547–5569. PMLR.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Elias Frantar and Dan Alistarh. 2023. Qmoe: Practical sub-1-bit compression of trillion-parameter models. *Preprint*, arXiv:2310.16795.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers. *Preprint*, arXiv:2210.17323.

Fei Gao. 2023. Density-based approach for fuzzy rule interpolation. *Applied Soft Computing*, 143:110402.

Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. 2024a. Harder task needs more experts: Dynamic routing in MoE models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12883–12895, Bangkok, Thailand. Association for Computational Linguistics.

Xijie Huang, Zechun Liu, Shih-Yang Liu, and Kwang-Ting Cheng. 2024b. RoLoRA: Fine-tuning rotated outlier-free LLMs for effective weight-activation quantization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7563–7576, Miami, Florida, USA. Association for Computational Linguistics.

Changhong Jiang, Changjing Shang, and Qiang Shen. 2025. Reinforcing tsk model interpolation with rule weight adjustment via location view analysis. *IEEE Transactions on Fuzzy Systems*, 33(2):580–592.

Szilveszter Kovács and László T Kóczy. 1997. Approximate fuzzy reasoning based on interpolation in the vague environment of the fuzzy rule base as a practical alternative of the classical cri. In *Proceedings of the 7th International Fuzzy Systems Association World Congress, Prague, Czech Republic*, pages 144–149.

LászlóT. Kóczy and Kaoru Hirota. 1993a. Approximate reasoning by linear rule interpolation and general approximation. *International Journal of Approximate Reasoning*, 9(3):197–225.

LászlóT. Kóczy and Kaoru Hirota. 1993b. Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases. *Information Sciences*, 71(1):169–201.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *Preprint*, arXiv:2006.16668.

Baohao Liao, Christian Herold, Shahram Khadivi, and Christof Monz. 2024. ApiQ: Finetuning of 2-bit quantized large language model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20996–21020, Miami, Florida, USA. Association for Computational Linguistics.

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. In *Proceedings of Machine Learning and Systems*, volume 6, pages 87–100.

Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. 2025. A closer look into mixture-of-experts in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4427–4447, Albuquerque, New Mexico. Association for Computational Linguistics.

Ang Lv, Jin Ma, Yiyuan Ma, and Siyuan Qiao. 2026. Coupling experts and routers in mixture-of-experts via an auxiliary loss. *Preprint*, arXiv:2512.23447.

Yuexiao Ma, Taisong Jin, Xiawu Zheng, Yan Wang, Huixia Li, Yongjian Wu, Guannan Jiang, Wei Zhang, and Rongrong Ji. 2023a. Ompq: Orthogonal mixed precision quantization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(7):9029–9037.

Yuexiao Ma, Huixia Li, Xiawu Zheng, Feng Ling, Xuefeng Xiao, Rui Wang, Shilei Wen, Fei Chao, and Rongrong Ji. 2024a. Affinequant: Affine transformation quantization for large language models. *Preprint*, arXiv:2403.12544.

Yuexiao Ma, Huixia Li, Xiawu Zheng, Feng Ling, Xuefeng Xiao, Rui Wang, Shilei Wen, Fei Chao, and Rongrong Ji. 2024b. Outlier-aware slicing for post-training quantization in vision transformer. In *Forty-first International Conference on Machine Learning*.

Yuexiao Ma, Huixia Li, Xiawu Zheng, Xuefeng Xiao, Rui Wang, Shilei Wen, Xin Pan, Fei Chao, and Rongrong Ji. 2023b. Solving oscillation problem in post-training quantization through a theoretical perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7950–7959.

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

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. 2020. Up or down? Adaptive rounding for post-training quantization. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 7197–7206. PMLR.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernandez. 2016. The lambada dataset: Word prediction requiring a broad discourse context. In *ACL*, pages 1525–1534.

Irina Proskurina, Luc Brun, Guillaume Metzler, and Julien Velcin. 2024. When quantization affects confidence of large language models? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1918–1928, Mexico City, Mexico. Association for Computational Linguistics.

Zihan Qiu, Zeyu Huang, Bo Zheng, Kaiyue Wen, Zekun Wang, Rui Men, Ivan Titov, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. Demons in the detail: On implementing load balancing loss for training specialized mixture-of-expert models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5005–5018, Vienna, Austria. Association for Computational Linguistics.

Yanpeng Qu, Jiaxing Wu, Zhanwen Wu, and Longzhi Yang. 2024. Fuzzy rule interpolation with a general representation of fuzzy sets. In *2024 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8.

Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. 2022. DeepSpeed-MoE: Advancing mixture-of-experts inference and training to power next-generation AI scale. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 18332–18346. PMLR.

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

Wei Tao, Haocheng Lu, Xiaoyang Qu, Bin Zhang, Kai Lu, Jiguang Wan, and Jianzong Wang. 2025. MoQAE: Mixed-precision quantization for long-context LLM inference via mixture of quantization-aware experts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10810–10820, Vienna, Austria. Association for Computational Linguistics.

An Wang, Xingwu Sun, Ruobing Xie, Shuaipeng Li, Jiaqi Zhu, Zhen Yang, Pinxue Zhao, Weidong Han, Zhanhui Kang, Di Wang, Naoaki Okazaki, and Cheng-zhong Xu. 2025. HMoE: Heterogeneous mixture of experts for language modeling. In *EMNLP*, pages 21943–21957, Suzhou, China. Association for Computational Linguistics.

Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. 2019. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Zhanhao Xie, Yuexiao Ma, Xiawu Zheng, Fei Chao, Wanchen Sui, Yong Li, Shen Li, and Rongrong Ji. 2025. Automated fine-grained mixture-of-experts quantization. In *Findings of ACL*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Jiaqi Zhao, Miao Zhang, Chao Zeng, Ming Wang, Xuebo Liu, and Liqiang Nie. 2024. LRQuant: Learnable and robust post-training quantization for large language models. In *ACL*, pages 2240–2255.

A Interpolation Variables Definition

Table 7 details the input features used for the Quantization Error and Runtime Latency interpolation models in FRI-MxMoE. We distinguish between structural features (Layer Type), configuration parameters (Bit-width, Tiling), and scale-dependent features (Problem Size, FLOPs) to comprehensively capture the heterogeneous characteristics of MoE experts. These features were selected based on their high correlation with quantization sensitivity and hardware execution efficiency, as verified in preliminary experiments. The feature set is designed to be extensible, allowing for the inclusion of additional hardware-specific counters if available.

B Overall Algorithm

Algorithm 1 provides the detailed pseudocode for the FRI-MxMoE framework. The process is divided into three main stages:

(1) Sparse Rule Base Construction: We construct a sparse rule base by profiling a limited number of anchor experts (Eq. (3)). This significantly reduces the calibration overhead from linear $O(E)$ to constant $O(1)$.

(2) FRI Interpolation: We use fuzzy rule interpolation (Eq. (4)–(5)) to predict the quantization error $\widehat{\mathcal{L}}$ and runtime cost $\widehat{\mathcal{C}}$ for all possible configurations in the full search space, handling the heterogeneity of unmeasured experts.

(3) Lagrangian Allocation: Finally, we solve the budget-constrained mixed-precision allocation problem using a Lagrangian relaxation approach (Eq. (6)–(8)), which efficiently finds the optimal bit-width assignment via binary search on the Lagrange multiplier λ.

C Rule Base Size Analysis

We compare the size of the search space in MxMoE with the effective rule base size in FRI-MxMoE across different models. As shown in Figure 3, while the search space of MxMoE grows linearly with the number of experts (reaching over 150k atomic configurations for DeepSeek-V2), the rule base size of FRI-MxMoE remains relatively constant (around 4k–10k) once the anchor budget is fixed. This demonstrates the scalability of our approach. Furthermore, the rule base size in FRI-MxMoE is decoupled from the total number of experts, depending instead on the diversity of expert types (e.g., parameter scales, activation patterns). This property makes FRI-MxMoE particularly advantageous for future massive MoE models with thousands of experts.

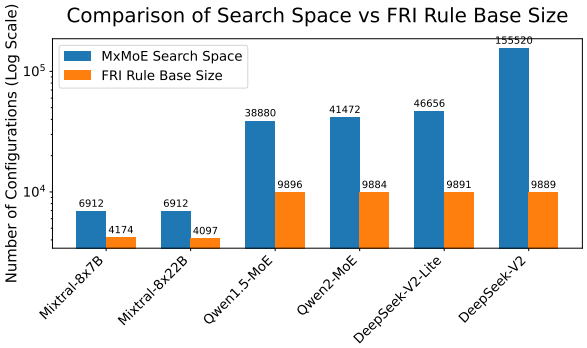

Figure 3: Comparison of Search Space Size (MxMoE) and Rule Base Size (FRI-MxMoE) across different models. Note the log scale on the y-axis.

In our Lagrangian-based resource allocation (Section 3.2.5), the hyperparameter λ (Lagrange multiplier) plays a pivotal role in balancing quantization error and runtime cost. The optimization objective is formulated as:

$$\min_{\mathbf{b}} \left(\sum_i f_i \cdot \hat{\mathcal{L}}_i(b_i) + \lambda \sum_i \hat{\mathcal{C}}_i(b_i) \right). \quad (9)$$

As λ increases, the penalty for runtime cost $\hat{\mathcal{C}}$ grows, forcing the solver to select bit-width configurations with lower cost (and typically lower precision). In contrast, a smaller λ relaxes the cost constraint, allowing the model to retain higher precision.

D Impact of Lagrangian Multiplier λ

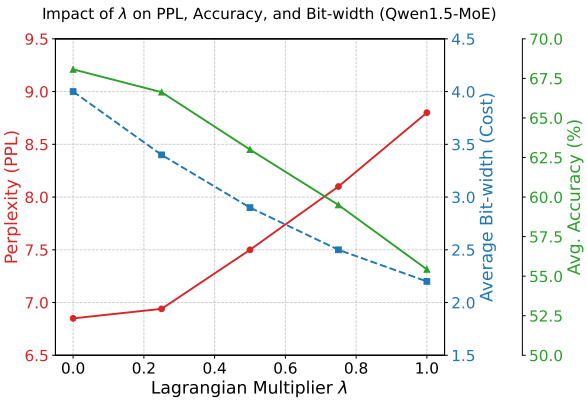

Figure 4: Impact of Lagrangian Multiplier λ on Perplexity (PPL), Avg. Accuracy, and Average Bit-width for Qwen1.5-MoE. $\lambda = 0.25$ provides a balanced trade-off.

Table 7: Definition of linguistic variables (input features) used in FRI-MxMoE's interpolation models. **Type**: D (Discrete/Categorical), C (Continuous).

Model	Variable	Type	Description
Quantization Error (Interpolating $\hat{\mathcal{L}}$)	Structural Type (t)	D	Sub-layer type: $\{\text{Gate}, \text{Up}, \text{Down}\}$, reflecting functional role.
	Bit-width (b)	C	Quantization precision, $b \in [2, 16]$, directly controlling noise.
	Scale Features	C	M, N, K, FLOPs distinguishing large vs. small layers.
	AI Proxy	C	Derived feature FLOPs$/(K \cdot N \cdot b)$, capturing arithmetic intensity.
	Activation Variance	C	Variance of input activations, indicating outlier sensitivity.
Runtime Latency (Interpolating $\hat{\mathcal{C}}$)	Tiling Config	C	Tile sizes BM, BN, BK determining register pressure.
	Pipeline Stage (S)	C	Number of pipeline stages (e.g., 2), affecting instruction overlap.
	Problem Size	C	M, N, K determining total workload and tail effects.
	Precision Config	C	Weight (W_{bits}) and Activation (A_{bits}) bit-widths affecting memory traffic.
	Fusion Status (F)	D	Boolean flag for operator fusion, impacting launch overhead.
	Thread Block	C	Number of threads per block, affecting occupancy.

Table 8: Sensitivity to feature engineering (group ablation) on Qwen1.5-MoE under W3.25-A16. We keep the Lagrangian allocator fixed and ablate groups of fuzzy inputs for the *error* FRI module (latency module uses the full feature set). "Avg." is the average accuracy over seven zero-shot tasks; PPL is evaluated on Wikitext-2.

Error-FRI Feature Set	Bit	Act. Stats	Weight Scale	Avg. Acc (%)	PPL
Full (FRI-MxMoE, paper)	✓	✓	✓	66.63	6.94
w/o activation statistics	✓	–	✓	66.22	7.06
w/o weight-scale features	✓	✓	–	66.38	6.98
Bit-only (no stats/scale)	✓	–	–	65.77	7.12
MxMoE	–	–	–	66.49	7.02

Table 9: Prediction error vs. rule-base size on Qwen1.5-MoE. We compare our depth-adaptive stratified sampling with random sampling under the same rule budget.

Sampling	Rule Base	MAE ↓	RMSE ↓	Spearman ρ ↑
Stratified	1k	0.182	0.241	0.61
Stratified	2.5k	0.131	0.178	0.69
Stratified	5k	0.096	0.132	0.75
Stratified	7.5k	0.082	0.114	0.78
Stratified	10k	**0.076**	**0.106**	**0.80**
Random	1k	0.224	0.296	0.53
Random	2.5k	0.168	0.229	0.62
Random	5k	0.122	0.169	0.71
Random	7.5k	0.094	0.129	0.76
Random	10k	0.083	0.112	0.79

We analyze the impact of λ on model performance and average bit-width. Figure 4 quantitatively illustrates this trade-off for Qwen1.5-MoE. As λ increases from 0 to 1.0, the optimization places greater emphasis on efficiency (Cost minimization), leading to a reduction in Average Bit-width (blue dashed line). While a smaller λ yields lower Perplexity (red solid line) and higher Avg. Accuracy (green solid line), it comes at the expense of significantly increased computational cost and storage size (higher bit-width).

Specifically, at $\lambda = 0.25$ (which is the default value used in our experiments), we observe a "sweet spot" where the model achieves competitive performance (PPL $\approx$ 6.94, Avg. Acc $\approx$ 66.63%) with a moderate bit-width ($\approx$ 3.4 bits), balancing the trade-off effectively. Extremely low λ values (approaching 0) would maximize accuracy but result in bit-widths close to full precision, negating the benefits of quantization.

E Prediction Error vs. Rule-base Size

As Table 9 shows, the same trend appears across all budgets: increasing the rule-base size consistently reduces prediction error, but the marginal gain gradually diminishes once the anchor coverage becomes sufficiently dense. This pattern is expected for local interpolation: early anchors rapidly improve coverage of the heterogeneous expert-layer space, whereas later anchors mainly refine already well-covered regions and therefore bring smaller incremental benefits. In other words, the first few thousand rules are primarily spent on discovering the coarse geometry of the sensitivity manifold, while the last few thousand rules mostly sharpen local details that have much smaller impact on the final ranking. This is particularly important for our use case because mixed-precision allocation depends more on preserving the relative ordering of sensitive units than on driving every pointwise predic-

Table 10: Sensitivity to feature engineering (group ablation) across MoE LLMs under W3.25-A16. We keep the Lagrangian allocator fixed and ablate groups of fuzzy inputs for the *error* FRI module (latency module uses the full feature set). "Avg." is the average accuracy over seven zero-shot tasks; PPL is evaluated on Wikitext-2. Δ is computed w.r.t. the Full FRI-MxMoE setting within each model.

Model	Error-FRI Feature Set	Avg. Acc (%)	PPL	ΔAvg	ΔPPL
DeepSeekV2-Lite	Full (FRI-MxMoE)	70.41	5.96	+0.00	+0.00
	w/o activation statistics	69.92	6.09	-0.49	+0.13
	w/o weight-scale features	70.15	6.01	-0.26	+0.05
	Bit-only (no stats/scale)	69.54	6.16	-0.87	+0.20
	MxMoE (ILP, paper)	69.37	6.08	-1.04	+0.12
Qwen1.5-MoE	Full (FRI-MxMoE)	66.63	6.94	+0.00	+0.00
	w/o activation statistics	66.22	7.06	-0.41	+0.12
	w/o weight-scale features	66.38	6.98	-0.25	+0.04
	Bit-only (no stats/scale)	65.77	7.12	-0.86	+0.18
	MxMoE (ILP, paper)	66.49	7.02	-0.14	+0.08
Qwen2-MoE	Full (FRI-MxMoE)	71.14	6.23	+0.00	+0.00
	w/o activation statistics	70.79	6.34	-0.35	+0.11
	w/o weight-scale features	70.98	6.27	-0.16	+0.04
	Bit-only (no stats/scale)	70.42	6.39	-0.72	+0.16
	MxMoE (ILP, paper)	71.05	6.18	-0.09	-0.05
Mixtral-8×7B	Full (FRI-MxMoE)	77.55	4.11	+0.00	+0.00
	w/o activation statistics	77.21	4.19	-0.34	+0.08
	w/o weight-scale features	77.36	4.14	-0.19	+0.03
	Bit-only (no stats/scale)	76.88	4.24	-0.67	+0.13
	MxMoE (ILP, paper)	77.49	4.15	-0.06	+0.04

Table 11: Sensitivity to alternative but semantically equivalent statistics on Qwen1.5-MoE (W3.25-A16). We replace the activation-statistics group in Error-FRI while keeping other inputs and the allocator unchanged.

Activation-statistics Variant (Error-FRI)	Definition	Avg. Acc (%)	PPL
Variance (paper default)	$\mathrm{Var}(a)$	66.63	6.94
Mean-abs activation	$\mathbb{E}[\lvert a\rvert]$	66.02	6.99
Outlier ratio	$\Pr(\lvert a\rvert > \tau)$ (fixed τ)	66.34	6.96
Kurtosis (tail heaviness)	$\mathrm{Kurt}(a)$	65.86	7.05

tion error to its absolute minimum. The steady improvement in Spearman correlation therefore matters as much as the MAE/RMSE trend: once the correlation is already high, the allocator can usually recover a strong bit-width assignment even if small residual prediction noise remains.

At every budget level, our depth-adaptive stratified sampling also remains more sample-efficient than random sampling, indicating that the quality of anchor placement matters as much as the raw number of anchors. This result is consistent with the structure of MoE models: sensitivity is not uniformly distributed across depth, expert role, and sub-layer type, so random sampling tends to waste budget on redundant regions while missing structurally important ones. By contrast, stratified sampling deliberately allocates anchors to regions where sensitivity heterogeneity is larger, which improves both absolute prediction error and the ordering fidelity needed for downstream allocation. The comparison therefore supports the design principle behind FRI-MxMoE: when profiling is sparse, careful coverage of the feature space is more valuable than naively increasing the number of anchors without structure.

From a practical perspective, these results support a simple deployment recipe: one can start from a fixed anchor schedule and use a modest held-out validation split to check whether the current rule budget is already sufficient, only increasing the budget when the interpolation error remains above an acceptable threshold for a new model or hardware setting.

F Stability Properties of IDW-FRI

For completeness, we restate the three practical stability properties used in Section 3.2.3. Let $\alpha_i = w_i / \sum_j w_j$ denote the normalized IDW weights for the K nearest rules. Then $\alpha_i \geq 0$ and $\sum_i \alpha_i = 1$, so the prediction can be written as a convex

combination:

$$\hat{y} = \sum_i \alpha_i y^{(i)}. \tag{10}$$

This immediately implies **boundedness**:

$$\min_i y^{(i)} \le \hat{y} \le \max_i y^{(i)}. \tag{11}$$

If the anchor outputs are corrupted by bounded perturbations $|\delta_i| \le \eta$, then the perturbed prediction satisfies

$$|\delta\hat{y}| = \left|\sum_i \alpha_i \delta_i\right| \le \sum_i \alpha_i |\delta_i| \le \eta, \tag{12}$$

which gives **noise robustness** with respect to local anchor noise. This property matters because anchor profiling is itself imperfect: the measured loss or latency for a profiled configuration can fluctuate with calibration data, quantization noise, or system-level measurement variance. The inequality above shows that such local anchor noise cannot be amplified by the interpolation rule beyond the worst perturbation already present in the selected neighbourhood. As a result, IDW-FRI behaves conservatively: it may inherit local uncertainty, but it does not create additional instability on top of that uncertainty.

Finally, when $\epsilon > 0$ and all continuous features are normalized, both $D(\mathbf{x}_q, R_i)$ and $w_i = 1/(D(\mathbf{x}_q, R_i) + \epsilon)$ are smooth for queries away from exact rule collisions. Because sums, products, and quotients of smooth functions remain locally Lipschitz when the denominator stays positive, $\hat{y}$ is locally Lipschitz in $\mathbf{x}_q$. This continuity property is important for optimization: small changes in bit-width or feature values should not induce abrupt "cliffs" in the predicted sensitivity surface, because such discontinuities would make the downstream allocation problem much harder and more brittle. Taken together, boundedness, local noise robustness, and continuity explain why IDW-FRI yields a smooth and physically plausible sensitivity manifold that is suitable for Lagrangian-based allocation. These results do not provide a global task-level generalization bound, but they formalize why IDW-FRI avoids abrupt prediction cliffs and why isolated noisy anchors cannot dominate the interpolated sensitivity surface.

G Additional Analyses

This subsection collects the additional empirical analyses introduced in the rebuttal. We focus on whether FRI-MxMoE depends heavily on a particular feature design and whether its performance remains stable when semantically similar statistics are substituted.

G.1 Feature-group Ablation Across Models

We first evaluate the contribution of each feature group used by the error-FRI module while keeping the latency module and allocator fixed. Table 8 provides a compact single-model view on Qwen1.5-MoE, and Table 10 extends the same ablation to four MoE families. This ablation is intended to separate genuine semantic utility from accidental gains caused by a particular hand-crafted feature recipe. The goal is not to show that every feature is indispensable, but to verify that the method degrades gracefully when one semantic group is removed. On Qwen1.5-MoE, removing activation statistics or weight-scale features produces only moderate degradation, while the bit-only variant remains competitive but consistently worse than the full model. Across all four architectures, the same trend repeats: activation statistics are the most useful auxiliary cue, weight-scale features also help, and the full feature set is the most reliable overall. This suggests that FRI-MxMoE benefits from physically meaningful features, but does not hinge on a brittle, model-specific recipe.

G.2 Alternative Activation Statistics

We next test whether the method depends specifically on variance as the activation-statistics descriptor. Table 11 replaces variance with semantically related alternatives while keeping the remaining inputs and the allocator unchanged. This isolates whether FRI-MxMoE relies on one hand-crafted statistic or on the broader notion of activation magnitude / outlierness. The results remain close to the default configuration: mean-absolute activation and outlier ratio both preserve nearly the same quality, while kurtosis is slightly weaker but still workable. This indicates that the proposed interpolation framework depends more on capturing the semantic role of activation scale and outliers than on any single handcrafted statistic. In practice, this makes the feature design easier to adapt to new model families where the most informative activation descriptor may differ.

Algorithm 1 FRI-MxMoE: Sparse FRI Profiling with Lagrangian-based Resource Allocation

Require:

- Candidate bit-width set $\mathcal{B}$ (e.g., $\{2, 3, 4, \ldots, 16\}$).
- Anchor budget N_s (number of sampled experts for profiling).
- Interpolation hyperparameters: KNN size K, IDW smoothing ϵ.
- Resource budget C_{target} (e.g., target latency or model size).
- Expert activation frequencies $\{f_i\}$ from calibration data.

Ensure: Optimal bit-width assignment $b^* = \{b_i^*\}_i$ for all quantization units.

Phase 1: Sparse Rule Base Construction (Profiling)

Initialize empty anchor sets $S_L \leftarrow \emptyset$ (for Error), $S_C \leftarrow \emptyset$ (for Cost).
Sample N_s anchor configurations $\{x^{(i)}\}_{i=1}^{N_s}$ using Depth-Adaptive Stratified Sampling to cover diverse expert characteristics.
for $i = 1$ TO N_s **do**
    Profile anchor $x^{(i)}$ on calibration dataset:
        Measure Quantization Error $L^{(i)}$ (e.g., MSE or PPL degradation).
        Measure Runtime Cost $C^{(i)}$ (e.g., latency in ms).
    Add to datasets: $S_L \leftarrow S_L \cup \{(x^{(i)}, L^{(i)})\}$; $S_C \leftarrow S_C \cup \{(x^{(i)}, C^{(i)})\}$.
end for
Fuzzification: Convert each sample $(x^{(i)}, y^{(i)})$ into a fuzzy rule R_i:
    `IF` x_1 is $A_1^{(i)} \wedge \cdots \wedge x_d$ is $A_d^{(i)}$ `THEN` y is $y^{(i)}$.
Construct sparse rule bases $\mathcal{R}_L$ from S_L and $\mathcal{R}_C$ from S_C.

Phase 2: Full-Space Performance Prediction (FRI)

function IDW_INTERPOLATE($x_q, \mathcal{R}, K, \epsilon$)
    Compute heterogeneous distance $D(x_q, R_j)$ for all rules $R_j \in \mathcal{R}$ (Eq. 4).
    Identify set $\mathcal{N}_K(x_q)$ of K nearest rules with smallest distances.
    Compute interpolation weights and weighted average (Eq. 5).
    return Predicted value $\widehat{y}$.
end function
Predict for all quantization units i (experts/layers) and all bit-widths $b \in \mathcal{B}$:
for all unit i, bit-width b **do**
    Construct query feature vector $x_q \leftarrow$ ExtractFeatures(i, b).
    Predict Error: $\widehat{L}_i(b) \leftarrow$ IDW_INTERPOLATE$(x_q, \mathcal{R}_L, K, \epsilon)$.
    Predict Cost: $\widehat{C}_i(b) \leftarrow$ IDW_INTERPOLATE$(x_q, \mathcal{R}_C, K, \epsilon)$.
end for

Phase 3: Lagrangian-based Resource Allocation

Objective: Solve budget-constrained allocation problem (Eq. 6).
Initialize Lagrange multiplier range: $\lambda_{\min} \leftarrow 0$, $\lambda_{\max} \leftarrow \lambda_{\text{large}}$.
repeat
    Update $\lambda \leftarrow (\lambda_{\min} + \lambda_{\max})/2$.
    Local Optimization: For each unit i, select best bit-width independently (Eq. 7):
        $b_i^*(\lambda) \leftarrow$ optimal choice for current λ.
    Calculate total cost $C_{\text{total}}(\lambda)$ (Eq. 8).
    if $C_{\text{total}}(\lambda) \leq C_{\text{target}}$ **then**
        $\lambda_{\max} \leftarrow \lambda$ ▷ Under budget: try smaller λ to reduce error.
    else
        $\lambda_{\min} \leftarrow \lambda$ ▷ Over budget: increase λ to penalize cost.
    end if
until $|C_{\text{total}}(\lambda) - C_{\text{target}}| \leq \delta$ or iter limit reached
return Final assignment $b^* = \{b_i^*(\lambda)\}_i$.