BITSMOE: COST-AWARE BIT ALLOCATION IN SPECTRAL SPACE FOR MOE LLM QUANTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

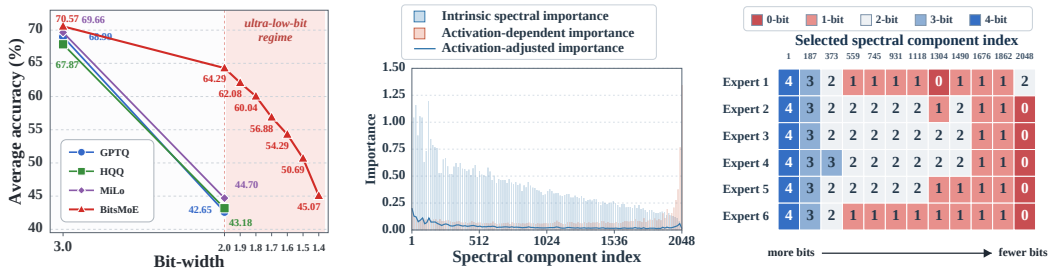
Mixture-of-Experts (MoE) large language models incur substantial memory costs due to their large expert parameter counts. Mixed-precision quantization reduces these costs by allocating different bit-widths to experts or linear blocks according to their importance. However, assigning a single precision within each expert or linear block overlooks its internal structural heterogeneity. This limitation motivates two key questions: **(1)** how to define a fine-grained unit for quantization within a linear transformation; and **(2)** how to characterize the quantization cost of each unit under actual activation patterns and different bit-widths. To address these two questions, we propose BITSMOE, a cost-aware mixed-precision quantization framework built on two complementary techniques: **(1)** Shared-basis Spectral Decomposition (SSD) separates expert weights into a shared basis and expert-specific spectral components, defining structural quantization units while exploiting cross-expert redundancy. **(2)** Factorized Quantization Cost Modeling (FQCM) estimates component-wise costs from output reconstruction loss by combining intrinsic spectral importance, activation-dependent importance, and bit-width-dependent distortion. Using these component-wise costs, we formulate bit allocation as an integer linear program (ILP) that minimizes total modeled quantization cost under a fixed memory budget. On Qwen3-30B-A3B at 2-bit, BITSMOE achieves 64.29% average accuracy over seven downstream tasks, outperforming the evaluated MoE-specific methods, including those using ILP-based bit allocation, and exceeding GEMQ by 2.80 percentage points. Under the same setting, it achieves a $16.47\times$ end-to-end offline quantization speedup over GEMQ. It also achieves up to $6.46\times$ the decode throughput of GPTQ.

1 INTRODUCTION

Mixture-of-Experts (MoE) (Cai et al., 2025) has emerged as one of the most effective paradigms for scaling large language models (LLMs). By leveraging dynamic routing mechanisms and scalable capabilities, MoE models have demonstrated superior performance and achieved state-of-the-art results (Jiang et al., 2024; DeepSeek-AI, 2024). However, the growth in expert count substantially increases parameter storage and memory bandwidth requirements. For instance, Qwen3-Next-80B-A3B (Qwen Team, 2025) activates only 3B parameters per token but has an FP16 GPU memory footprint of approximately 149 GB. These extreme resource requirements make deployment on memory-constrained GPUs largely infeasible.

To address these costs, post-training quantization (PTQ) has emerged as a promising solution for LLMs without the need for retraining (Lin et al., 2024; Xiao et al., 2023; Shao et al., 2024; Ashkboos et al., 2024). Typical PTQ methods treat all weight matrices uniformly, without accounting for the fact that MoE experts differ in their functional roles and contributions to model performance. Recent MoE-specific quantization methods (Duanmu et al., 2025; Huang et al., 2025) recognize such expert-level heterogeneity and tailor quantization configurations at the expert or linear-block level, thereby mitigating performance degradation at low bit-widths. However, assigning a single quantization configuration to an expert or linear block overlooks its intrinsic structural heterogeneity. Specifically, the quantization sensitivity of a linear transformation is inherently non-uniform across directions and further varies with the activation patterns presented to it.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Ultra-low-bit accuracy. (b) Spectral importance. (c) Component-wise allocation.

Figure 1: (a) Average accuracy over seven downstream tasks under ultra-low-bit quantization on Qwen3-30B-A3B. (b) Spectral components exhibit highly non-uniform importance. (c) BITSMOE assigns heterogeneous bit-widths to individual spectral components under a fixed bit budget.

These observations raise two key issues: ❶ **What should be the fine-grained unit for quantization within a linear transformation?** Simply partitioning a weight matrix along channels or dimensions offers finer granularity but lacks structural meaning. Moreover, independent partitioning fails to exploit the substantial redundancy among experts within the same MoE layer (Liu et al., 2025; Li et al., 2025). An effective quantization unit should therefore capture both the intrinsic structure and the shared structure across experts. ❷ **How should the quantization cost of each unit be characterized under actual activation patterns and different bit-widths?** Quantization cost depends not only on actual activation patterns, but also on the intrinsic sensitivity of each unit and the perturbation introduced by the quantizer. These factors jointly determine the impact of assigning a particular bit-width. A meaningful cost model should therefore couple intrinsic structure, activation-dependent importance, and quantization distortion to guide mixed-precision allocation.

MoE-SVD (Li et al., 2025) reduces redundancy among experts by sharing an SVD-derived basis. Delta-CoMe (Ping et al., 2024) uses singular vectors as quantization units and assigns them different bit-widths to compress delta weights. These ideas provide a starting point for the problem ❶. However, a fixed bit-allocation strategy based solely on singular-value ordering does not explicitly account for differences in quantization cost across units. This leaves problem ❷ unresolved: how to combine intrinsic sensitivity, activation-dependent importance, and bit-width-dependent quantization distortion into a quantization cost model for each unit.

To address these issues, we propose BITSMOE, which is built on two efficient innovations:❶ **Shared-basis Spectral Decomposition (SSD)**. First, we decompose the expert weights within each MoE layer into a shared basis and expert-specific spectral components. The shared basis captures common structure across experts, while each spectral component serves as a fine-grained structural unit for quantization. ❷ **Factorized Quantization Cost Modeling (FQCM)**. Second, we model the quantization cost of each component by jointly considering its intrinsic spectral importance, activation-dependent importance, and quantization distortion. Based on this cost model, we formulate bit allocation as an integer linear program (ILP) that assigns a bit-width to each spectral component under a fixed bit budget. As illustrated in Figs. 1b and 1c, spectral components vary substantially in spectral importance, and BITSMOE assigns heterogeneous bit-widths accordingly. Extensive experiments demonstrate that BITSMOE achieves strong accuracy across a range of bit budgets in the ultra-low-bit regime (Fig. 1a).

Our main contributions are summarized as follows:

- We introduce **Shared-basis Spectral Decomposition (SSD)** to define structurally meaningful, fine-grained quantization units within MoE experts. SSD separates shared structure across experts from expert-specific spectral components, capturing cross-expert redundancy while exposing the intrinsic heterogeneity within each expert.
- We derive **Factorized Quantization Cost Modeling (FQCM)** from the output reconstruction loss, decomposing component-wise quantization cost into intrinsic spectral importance, activation-dependent importance, and bit-width-dependent distortion. FQCM estimates the impact of each precision assignment under observed activation patterns, providing a cost criterion beyond singular-value ordering.

- Building on SSD and FQCM, we formulate MoE quantization as a **fine-grained bit-budget allocation problem**. The resulting ILP minimizes the total modeled quantization cost under a fixed memory budget, enabling cost-aware precision allocation that selectively preserves critical expert-specific components.
- Experiments demonstrate the accuracy, offline quantization efficiency, and inference efficiency of **BITSMOE**. At 2-bit precision on Qwen3-30B-A3B, BITSMOE achieves 64.29% average accuracy over seven downstream tasks, outperforming the evaluated MoE-specific methods, including those using ILP-based bit allocation, and exceeding GEMQ by 2.80 percentage points. Under the same setting, it achieves a $16.47\times$ end-to-end offline quantization speedup over GEMQ. It also delivers up to $6.46\times$ the decode throughput of GPTQ.

2 RELATED WORK

Mixture-of-Experts Large Language Models MoE architectures have become widely adopted in recent LLMs (Jiang et al., 2024; Liu et al., 2024; Xue et al., 2024; Muennighoff et al., 2025). By partitioning the network into multiple experts and routing each input to a sparse subset, MoE reduces per-token computation while improving scalability (Shazeer et al., 2017; Fedus et al., 2022). Despite these advantages, MoE LLMs still suffer from a large parameter footprint due to expert replication (He et al., 2023). Moreover, unbalanced routing induces expert-level redundancy and highly skewed expert utilization, which creates substantial disparities in expert importance and complicates effective compression (Lu et al., 2024).

MoE LLM Compression and Pruning SVD-based low-rank decomposition has been widely used as a structured compression tool for dense LLMs (Hsu et al., 2022; CHen et al., 2021; Yuan et al., 2025; Wang et al., 2025). For MoE LLMs, recent methods further exploit expert-level redundancy through pruning and structured decomposition. MoE-I^2 (Yang et al., 2024) combines non-uniform inter-expert pruning with importance-aware intra-expert low-rank decomposition to compress MoE LLMs in a task-agnostic framework. MoE-SVD (Li et al., 2025) selectively decomposes less sensitive expert layers and reduces cross-expert redundancy through frequency-guided V-matrix sharing and U-matrix trimming. D^2-MoE (Gu et al., 2025) decomposes expert weights into a Fisher-weighted shared base and expert-specific delta weights, where the shared base is compressed via semi-dynamic pruning and the delta weights are compressed via truncation-aware SVD.

Post-Training Quantization for MoE LLMs Post-training quantization (PTQ) has become a widely used paradigm for compressing LLMs without retraining. In this work, we focus on scalar weight quantization, a representative PTQ family that has been extensively studied for LLM compression. Among these methods, GPTQ (Frantar et al., 2022) uses Hessian-based error compensation for sequential weight quantization, while HQQ (Badri & Shaji, 2023) formulates low-bit quantization as a calibration-free half-quadratic optimization problem. For MoE LLMs, MoEQuant (Chen et al., 2025) improves PTQ by constructing expert-balanced calibration samples and incorporating token expert affinities into the quantization process. MiLo (Huang et al., 2025) augments extremely quantized MoE models with adaptive low-rank compensators and efficient INT3 kernels to recover accuracy while improving inference efficiency. MxMoE (Duanmu et al., 2025) assigns bit-widths according to block sensitivity, expert activation patterns, and hardware constraints, and generates optimized Group GEMM kernels for efficient MoE inference. GEMQ (Deng et al., 2026) uses global linear programming to allocate expert bit-widths based on estimated task-loss increases and combines this allocation with router fine-tuning in a progressive quantization framework.

3 METHODOLOGY

3.1 OVERVIEW OF BITSMOE

Fig. 2 presents BITSMOE, which addresses the two issues identified in Sec. 1 through the following components. We first construct a shared spectral basis across experts and identify expert-specific spectral directions as quantization units (Sec. 3.2). We then model their quantization costs to guide bit allocation under a fixed budget (Sec. 3.3), and organize the quantized directions for packed GPU execution (Sec. 3.4). Detailed derivations are provided in App. C.

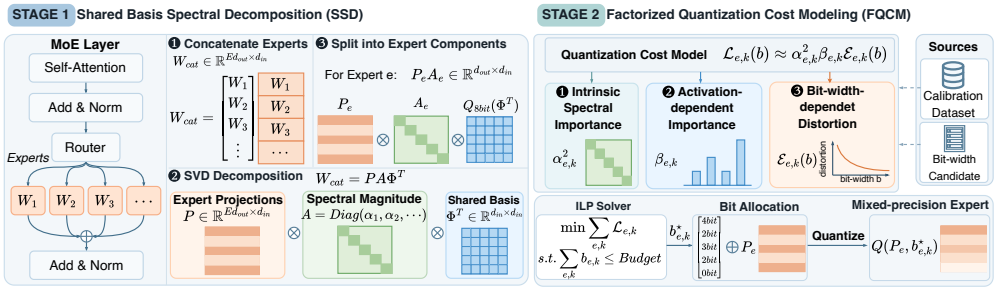


Figure 2: Overview of BITSMOE, a mixed-precision quantization framework for MoE LLMs.

3.2 SHARED-BASIS SPECTRAL DECOMPOSITION (SSD)

Within an MoE layer, expert weights operate on the same feature space and often exhibit substantial redundancy. Instead of decomposing each expert independently, we jointly decompose all experts to extract a shared spectral basis. We present a linear projection whose shared basis acts on its input space. App. C.1 provides the projection-specific formulations. Specifically, for expert weights $\{W_e\}_{e=1}^{E}$, where $W_e \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, we concatenate them and perform SVD:

$$W_{\text{cat}} := \begin{bmatrix} W_1 \\ \vdots \\ W_E \end{bmatrix} = U \mathbf{\Sigma} \mathbf{\Phi}^\top. \tag{1}$$

Here, $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n]$ denotes the shared spectral basis acting on the common feature space across experts within an MoE layer, where $n = \min(Ed_{\text{out}}, d_{\text{in}})$ is the number of spectral components. Partitioning $U\mathbf{\Sigma}$ into expert-specific spectral components yields

$$W_e = P_e A_e \mathbf{\Phi}^\top = \sum_{k=1}^{n} \boldsymbol{\alpha}_{e,k} p_{e,k} \boldsymbol{\phi}_k^\top, \qquad e = 1, \ldots, E, \tag{2}$$

where $p_{e,k}$ is the expert-specific spectral direction with $\|p_{e,k}\|_2 = 1$ and $\boldsymbol{\alpha}_{e,k}$ is the corresponding component magnitude. This representation decouples direction and magnitude within each spectral component $C_{e,k} = \boldsymbol{\alpha}_{e,k} p_{e,k} \boldsymbol{\phi}_k^\top$.

We define the spectral energy of each component as its squared Frobenius norm, which equals $\boldsymbol{\alpha}_{e,k}^2$:

$$\|C_{e,k}\|_F^2 = \text{Tr}(C_{e,k}^\top C_{e,k}) = \text{Tr}\left[(\boldsymbol{\alpha}_{e,k} \boldsymbol{\phi}_k p_{e,k}^\top) \boldsymbol{\alpha}_{e,k} p_{e,k} \boldsymbol{\phi}_k^\top\right] = \boldsymbol{\alpha}_{e,k}^2 \,\text{Tr}(\boldsymbol{\phi}_k p_{e,k}^\top p_{e,k} \boldsymbol{\phi}_k^\top) = \boldsymbol{\alpha}_{e,k}^2 \tag{3}$$

For an input activation matrix $X \in \mathbb{R}^{d_{\text{in}} \times t}$ with t tokens, the expert output can be expressed as

$$Y_e = W_e X = \sum_{k=1}^{n} \boldsymbol{\alpha}_{e,k} p_{e,k} (\boldsymbol{\phi}_k^\top X) = \sum_{k=1}^{n} \boldsymbol{\alpha}_{e,k} p_{e,k} h_k \tag{4}$$

Here, h_k denotes the projection of the input activations onto $\boldsymbol{\phi}_k$. Since $\boldsymbol{\phi}_k$ is shared across experts, each projected input h_k is associated with an expert-specific spectral direction $p_{e,k}$ and a corresponding magnitude $\boldsymbol{\alpha}_{e,k}$. We therefore use the expert-specific spectral direction $p_{e,k}$ of each spectral component as the quantization unit in BITSMOE. Unlike channel-wise or dimension-wise partitioning, this unit simultaneously captures the shared structure across experts and the intrinsic structure of each expert.

3.3 FACTORIZED QUANTIZATION COST MODELING (FQCM)

Although SSD identifies spectral components as fine-grained quantization units within each MoE layer, the remaining question is how to allocate bits among them under a fixed budget. To guide this allocation, we model the cost of assigning each component a particular bit-width by jointly considering its intrinsic spectral importance, activation-dependent importance, and bit-width-dependent

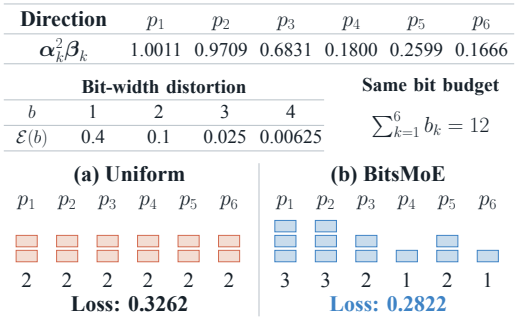


Figure 3: Uniform and BITSMOE bit allocations under the same budget.

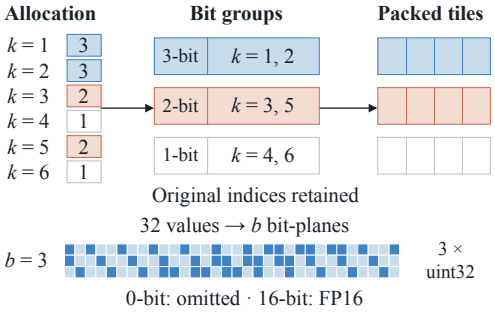


Figure 4: Components grouped by bit-width and packed into bit-planes.

distortion. The illustrative example in Fig. 3 shows how this combined cost guides allocation: under the same total budget of 12 bits, cost-aware allocation reduces the modeled loss from 0.3262 to 0.2822 compared with uniform allocation.

We first quantize the shared basis to 8-bit precision using GPTQ (Frantar et al., 2022) and denote the resulting basis by $\widehat{\mathbf{\Phi}}$. For the input-side form in Eq. (2), define $W_{e,q\phi} := P_e A_e \widehat{\mathbf{\Phi}}^\top$ and $\widehat{W}_e(b) := \widehat{P}_e(b) A_e \widehat{\mathbf{\Phi}}^\top$, with $\widehat{P}_e(b) := \big[Q_b(p_{e,1}), \ldots, Q_b(p_{e,n})\big]$. We measure the additional error introduced by quantizing the expert-specific spectral directions relative to $W_{e,q\phi}$. For per-token routing weights g_e, let $G_e := \mathrm{Diag}(g_e)$ and $H_{g,e} := X G_e X^\top$. Let $\boldsymbol{\varepsilon}_{e,k,b} = p_{e,k} - Q_b(p_{e,k})$ denote the error in each spectral direction. The weight error expands as

$$W_{e,q\phi} - \widehat{W}_e(b) = \big(P_e - \widehat{P}_e(b)\big) A_e \widehat{\mathbf{\Phi}}^\top = \sum_{k=1}^{n} \boldsymbol{\alpha}_{e,k} \boldsymbol{\varepsilon}_{e,k,b} \widehat{\boldsymbol{\phi}}_k^\top . \tag{5}$$

Under this metric, we define the expected routing-weighted output reconstruction loss from quantizing the spectral directions as

$$\begin{aligned} \mathcal{L}_{e,P}(b) &:= \mathbb{E}_\epsilon \left\| \big(W_{e,q\phi} - \widehat{W}_e(b)\big) X G_e^{1/2} \right\|_F^2 = \mathbb{E}_\epsilon \,\mathrm{Tr}\Big[\big(W_{e,q\phi} - \widehat{W}_e(b)\big) H_{g,e} \big(W_{e,q\phi} - \widehat{W}_e(b)\big)^\top \Big] \\ &= \sum_{k,l} \boldsymbol{\alpha}_{e,k} \boldsymbol{\alpha}_{e,l} \big(\widehat{\boldsymbol{\phi}}_k^\top H_{g,e} \widehat{\boldsymbol{\phi}}_l\big) \mathbb{E}_\epsilon \big[\boldsymbol{\varepsilon}_{e,l,b}^\top \boldsymbol{\varepsilon}_{e,k,b}\big] . \end{aligned} \tag{6}$$

$\mathbb{E}_\epsilon$ denotes expectation over the quantization errors. Under the diagonal component-error approximation (App. C.2), we approximate $\mathbb{E}_\epsilon[\boldsymbol{\varepsilon}_{e,l,b}^\top \boldsymbol{\varepsilon}_{e,k,b}]$ by zero for $k \neq l$. Define $\boldsymbol{\beta}_{e,k} := \widehat{\boldsymbol{\phi}}_k^\top H_{g,e} \widehat{\boldsymbol{\phi}}_k$ and $\mathcal{E}_{e,k,b} := \mathbb{E}_\epsilon \|\boldsymbol{\varepsilon}_{e,k,b}\|_2^2$. The remaining diagonal terms give

$$\mathcal{L}_{e,P}(b) \approx \sum_k \boldsymbol{\alpha}_{e,k}^2 \boldsymbol{\beta}_{e,k} \mathcal{E}_{e,k,b}. \tag{7}$$

This formulation decomposes the component-wise quantization cost of each spectral direction $p_{e,k}$ at bit-width b into three interpretable factors: **(1) Intrinsic spectral importance.** We define $\boldsymbol{\alpha}_{e,k}^2$ as the intrinsic spectral importance, which measures the calibration-agnostic strength of a spectral component in the expert weight transformation. **(2) Activation-dependent importance.** We define $\boldsymbol{\beta}_{e,k}$ as the activation-dependent importance, which captures how the calibration inputs and their activation patterns determine the importance of the spectral component. For bit allocation, we reuse $\boldsymbol{\beta}_{e,k}$ for its normalized diagonal approximation, as described in App. C.2. **(3) Bit-width-dependent distortion.** We define $\mathcal{E}_{e,k,b}$ as the bit-width-dependent distortion, which measures the quantization error incurred when assigning bit-width b to the spectral component. Under the high-resolution approximation for symmetric quantization, distortion decays proportionally to $\exp(-\lambda b)$ under 6-bit and 8-bit quantization, where $\lambda = 2 \ln 2$. At lower bit-widths, we use shared empirical coefficients for 2-bit, 3-bit, and 4-bit quantization and a direction-dependent sign-quantization error for 1-bit quantization, as detailed in App. C.3.

We model the component-wise quantization cost of assigning bit-width b to $p_{e,k}$ as

$$\mathcal{L}_{e,k}(b) \approx \underbrace{\boldsymbol{\alpha}_{e,k}^2}_{\substack{\text{Intrinsic spectral} \\ \text{importance}}} \cdot \underbrace{\boldsymbol{\beta}_{e,k}}_{\substack{\text{Activation-dependent} \\ \text{importance}}} \cdot \underbrace{\mathcal{E}_{e,k,b}}_{\substack{\text{Bit-width-dependent} \\ \text{distortion}}} . \tag{8}$$

3.4 Cost-Aware Bit Allocation and GPU Execution

Let $\ell \in [L]$ index the L MoE layers and $y_{\ell,e,k,b} \in \{0,1\}$ indicate whether component (ℓ, e, k) is assigned bit-width $b \in \mathcal{B} = \{16, 8, 6, 4, 3, 2, 1, 0\}$. Then we define the FQCM loss as:

$$\mathcal{L}_{\mathrm{FQCM}}(y) := \sum_{\ell=1}^{L} \sum_{e,k} \sum_{b \in \mathcal{B}} \mathcal{L}_{\ell,e,k}(b) y_{\ell,e,k,b}, \tag{9}$$

We minimize this global FQCM loss through a single ILP problem:

$$\text{MINIMIZE} \quad \mathcal{L}_{\mathrm{FQCM}}(y) \qquad \text{S.T.} \quad \sum_{b \in \mathcal{B}} y_{\ell,e,k,b} = 1, \quad \forall \ell, e, k, \quad \sum_{\ell,e,k} \sum_{b \in \mathcal{B}} b\, y_{\ell,e,k,b} \le B_{\mathrm{alloc}}(\mathfrak{b}_{\mathrm{eq}}). \tag{10}$$

Here, $B_{\mathrm{alloc}}(\mathfrak{b}_{\mathrm{eq}})$ denotes the global normalized bit budget for a target bit-width $\mathfrak{b}_{\mathrm{eq}}$. We solve this global ILP independently for each projection type in $\{\mathrm{gate_proj}, \mathrm{up_proj}, \mathrm{down_proj}\}$. The detailed derivation is provided in App. C.4.

Packed storage and GPU execution. We group spectral components by bit-width within each expert and preserve their original indices for indexed computation. Fig. 4 illustrates this organization. Each low-bit group uses a bit-plane layout: 32 b-bit values occupy b `uint32` words, one per bit position. Zero-bit directions are omitted, and 16-bit directions remain in FP16. During inference, kernels unpack and dequantize the stored directions on the fly to compute expert contributions without reconstructing dense expert weights. Within each MoE layer, we apply each shared-basis transform once per token across all routed experts. We use vLLM's Marlin W8A16 kernels (Frantar et al., 2025) for these shared-basis matrix multiplications.

4 Experiments

4.1 Experimental Setup

Models and Datasets. We evaluate three MoE LLMs: DeepSeek-V2-Lite (Liu et al., 2024), Qwen3-30B-A3B-Base (Yang et al., 2025) and Qwen3-Next-80B-A3B-Instruct (Qwen Team, 2025). Hereafter, the two Qwen checkpoints are abbreviated as Qwen3-30B-A3B and Qwen3-Next-80B-A3B. We evaluate the proposed BitsMoE on six zero-shot tasks, including ARC-Easy (Clark et al., 2018), HellaSwag (Zellers et al., 2019), MathQA (Amini et al., 2019), Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2021a), OpenBookQA (Mihaylov et al., 2018) and WinoGrande (Sakaguchi et al., 2021). Furthermore, we evaluate BitsMoE using Grade School Math 8K (GSM8K) (Cobbe et al., 2021), which assesses multi-step mathematical reasoning skills. We evaluate these seven tasks on NVIDIA A100 80GB PCIe GPUs using the open-source tool lm-evaluation-harness (version 0.4.9.1) (Sutawika et al., 2025), and the ILP problems are solved using the Gurobi Optimizer (Gurobi Optimization, LLC, 2024).

Baselines. Our baselines include representative LLM post-training quantization (PTQ) methods HQQ (Badri & Shaji, 2023) and GPTQ (Frantar et al., 2022). On Qwen3-30B-A3B, we select MoEQuant (Chen et al., 2025), MiLo (Huang et al., 2025), MxMoE (Duanmu et al., 2025), and GEMQ (Deng et al., 2026) as representative MoE-specific PTQ methods covering GPTQ-based quantization, low-rank decomposition for quantization-error compensation, hardware-aware ILP-based mixed-precision quantization, and global expert-level mixed-precision quantization, respectively. All methods quantize only MoE expert weights with group size 128.

4.2 Main Results

Tab. 1 shows that BitsMoE achieves the highest average accuracy at 2-bit on all three models while remaining competitive at 3-bit. On Qwen3-Next-80B-A3B, its 2-bit average accuracy of 70.70%

Table 1: Task-wise accuracy (%) of different methods on the seven downstream tasks.

Model	Method	Bits	Accuracy (%) ↑							
			HS	AE	MQ	MM	OB	WG	GS	Avg.
DEEPSEEK-V2-LITE	FP16	16	77.70	76.30	39.03	55.60	44.40	70.88	39.12	57.58
	HQQ	3	**76.83**	71.72	36.45	**53.16**	**44.40**	**70.88**	**32.15**	**55.08**
	GPTQ	3	75.88	73.48	37.29	50.92	44.20	69.30	30.40	54.50
	BITSMOE	3	74.65	**74.24**	**38.06**	52.86	41.60	70.01	29.34	54.39
	HQQ	2	67.79	61.45	29.25	**43.48**	38.00	63.85	11.98	45.11
	GPTQ	2	**68.81**	65.74	28.21	37.66	36.40	65.75	10.08	44.66
	BITSMOE	2.0	68.21	**70.54**	**32.70**	41.78	**39.60**	**67.01**	**12.21**	**47.44**
		1.9	66.33	69.15	31.76	42.00	38.80	67.32	11.14	46.64
		1.8	64.19	66.54	30.75	40.09	37.80	66.06	9.10	44.93
QWEN3-30B-A3B	FP16	16	81.35	79.80	60.03	78.77	45.00	72.85	83.47	71.61
	HQQ	3	78.55	76.56	49.01	75.37	44.40	71.67	79.53	67.87
	GPTQ	3	**79.92**	78.11	54.07	**75.83**	43.40	72.14	79.45	68.99
	BITSMOE	3	78.58	**79.88**	**56.21**	75.64	**46.80**	**74.27**	**82.64**	**70.57**
	HQQ	2	63.05	37.33	33.17	48.82	36.40	60.85	22.67	43.18
	GPTQ	2	70.16	59.97	24.89	39.17	39.40	60.62	4.32	42.65
	BITSMOE	2.0	**70.43**	**80.18**	**52.86**	**64.36**	**41.00**	**70.17**	**71.04**	**64.29**
		1.9	68.57	77.99	51.22	61.78	38.60	69.53	66.87	62.08
		1.8	66.21	76.14	48.41	60.40	37.00	67.40	64.75	60.04
		1.7	63.00	74.03	46.43	54.94	36.20	65.43	58.15	56.88
		1.6	60.57	73.70	43.45	48.18	35.00	65.04	54.06	54.29
QWEN3-NEXT-80B-A3B	FP16	16	82.72	84.64	63.85	84.53	44.20	76.80	77.18	73.42
	HQQ	3	**82.11**	83.50	61.17	**83.81**	**45.60**	**77.03**	76.57	72.83
	GPTQ	3	81.42	84.30	59.40	82.52	44.20	76.09	76.65	72.08
	BITSMOE	3	81.23	**85.35**	**63.25**	83.79	44.20	76.16	**77.03**	**73.00**
	HQQ	2	**78.73**	83.50	50.95	79.68	43.80	70.40	66.49	67.65
	GPTQ	2	70.24	**84.30**	27.47	54.63	38.60	65.59	18.35	51.31
	BITSMOE	2.0	78.05	83.92	**61.71**	**79.95**	**44.60**	**74.90**	**71.80**	**70.70**
		1.9	77.45	83.42	61.74	79.11	44.40	75.22	70.28	70.23
		1.8	76.37	83.04	59.80	77.93	43.80	74.59	69.07	69.23
		1.7	75.07	84.09	58.46	77.13	44.00	74.74	69.60	69.01
		1.6	73.39	83.63	56.52	75.97	42.60	74.19	66.79	67.58

exceeds HQQ by 3.05 percentage points. The advantage extends to mathematical reasoning below 2-bit precision. At 1.6-bit, Qwen3-30B-A3B retains 54.06% accuracy on GSM8K and 43.45% on MathQA, which exceed those of 2-bit HQQ by 31.39 and 10.28 percentage points, respectively. Mathematical reasoning is crucial for many real-world applications, so preserving this ability at ultra-low precision highlights the practical relevance of BITSMOE. Tab. 2 reports WikiText-2 (Merity et al., 2016) perplexity on the Qwen3-30B-A3B base model. BITSMOE achieves the lowest 2-bit perplexity of 9.12, which is 0.20 lower than GPTQ's 9.32.

As shown in Tab. 3, BITSMOE achieves the highest average accuracy of 64.29% at 2-bit on Qwen3-30B-A3B among MoE-specific methods, including those based on ILP allocation and low-rank error compensation. In particular, BITSMOE achieves 71.04% accuracy on GSM8K and surpasses GEMQ by 24.94 percentage points. This advantage further supports the practical value of BITSMOE for real-world applications that require multi-step mathematical reasoning.

4.3 ABLATION STUDY

Component importance and shared-basis precision. Tab. 4 evaluates SSD, FQCM, and shared-basis precision on Qwen3-30B-A3B at 2-bit. Non-Basis applies SVD independently to each expert, whereas SSD constructs a shared basis. Uniform assigns equal bit-widths to spectral components; FQCM adaptively allocates bits using the cost model. Uniform allocation achieves only 24.86%

Table 2: PPL on Qwen3-30B-A3B.

Method	Bits	W2 PPL↓
FP16	16	5.51
HQQ	2	12.52
GPTQ	2	9.32
MiLo	2	11.74
MoEQuant	2	12.04
BITSMOE	2	**9.12**

Table 3: Evaluation results on Qwen3-30B-A3B under 2-bit quantization using MoE-specific methods.

Method	Accuracy (%) ↑							
	HS	AE	MQ	MM	OB	WG	GS	Avg.
MiLo	62.82	58.71	31.83	44.51	35.40	59.98	19.64	44.70
MxMoE	76.91	59.30	22.81	48.45	40.40	71.27	1.90	45.86
MoEQuant	66.44	69.70	45.09	**70.02**	40.00	67.72	49.36	58.33
GEMQ	**78.68**	**81.44**	41.61	64.26	**44.00**	**74.35**	46.10	61.49
BITSMOE	70.43	80.18	**52.86**	64.36	41.00	70.17	**71.04**	**64.29**

Table 4: Average accuracy for ablations on Qwen3-30B-A3B at 2-bit. The Bits column specifies shared-basis storage precision.

Basis	Bits	Alloc.	α^2	β	G_e	Avg. Acc. (%) ↑
Non-Basis	–	Uniform	–	–	–	24.86
SSD	8-bit	Uniform	–	–	–	25.27
		FQCM	✗	✗	–	26.17
			✓	✗	–	26.48
			✗	✓	✗	47.20
			✗	✓	✓	57.92
			✓	✓	✗	60.45
	FP16	FQCM	✓	✓	✓	60.91
	4-bit		✓	✓	✓	62.78
	8-bit		✓	✓	✓	**64.29**

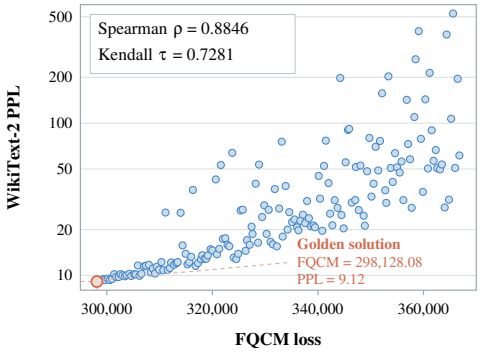


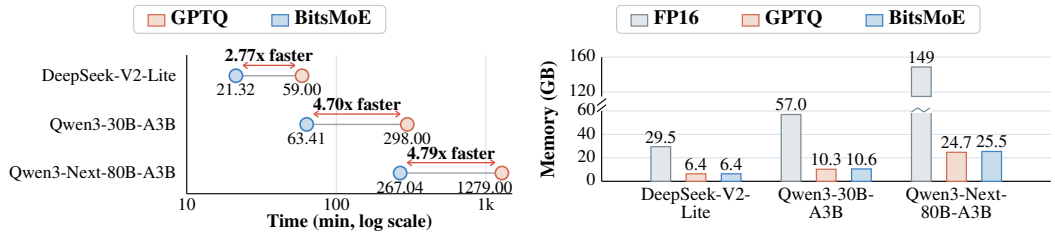
Figure 5: Correlation between FQCM loss and WikiText-2 PPL for 170 allocations at 2-bit on Qwen3-30B-A3B.

average accuracy with independent SVD and 25.27% with SSD, suggesting that basis sharing alone is insufficient without distinguishing component importance. With an 8-bit shared basis, using spectral energy $\boldsymbol{\alpha}^2$ as the sole importance signal achieves 26.48%: intrinsic spectral importance alone appears insufficient to guide component-wise bit allocation under actual activation patterns. Using only activation-dependent importance $\boldsymbol{\beta}$ instead accounts for observed activation patterns, raising accuracy to 47.20% without routing weights. Incorporating G_e accounts for expert-specific routing weights and further improves accuracy to 57.92%. Adding $\boldsymbol{\alpha}^2$ then reaches 64.29%, suggesting that spectral energy within experts complements their activation and routing information. Within the FQCM, 8-bit also outperforms FP16 and 4-bit shared bases. FP16 consumes more of the limited budget, leaving fewer bits for expert-specific components, whereas 4-bit errors may degrade the common transformations used by all experts within a MoE layer. We therefore adopt an 8-bit shared basis to balance storage cost and accuracy while supporting efficient computation with W8A16 kernels. The corresponding task-level results are reported in Tab. 10 in App. E.2.

FQCM as an allocation criterion. In Fig. 5, we randomly perturb the ILP bit allocation while preserving the 2-bit budget and evaluate WikiText-2 PPL for 170 allocations, including the ILP solution. Spearman's $\rho = 0.8846$ and Kendall's $\tau = 0.7281$ indicate a strong positive rank correlation between FQCM loss and PPL. PPL becomes more dispersed at higher FQCM losses, possibly reflecting nonlinear error propagation and limiting precise PPL prediction. In the low-loss region, FQCM loss and PPL exhibit a particularly strong positive rank correlation, supporting FQCM as an objective for ILP bit allocation. The ILP solution obtained by minimizing FQCM loss under the fixed budget lies in this region and achieves the lowest measured PPL among the evaluated allocations. This experiment validates the effectiveness of our FQCM for bit allocation.

4.4 EFFICIENCY ANALYSIS

Offline Quantization Overhead. On Qwen3-Next-80B-A3B, BITSMOE reduces end-to-end offline quantization time on an NVIDIA A100 PCIe (80 GB) GPU from GPTQ's 1,279 minutes to



(a) Quantization time ↓.

(b) Memory footprint ↓.

Figure 6: Quantization efficiency and memory footprint at 2-bit.

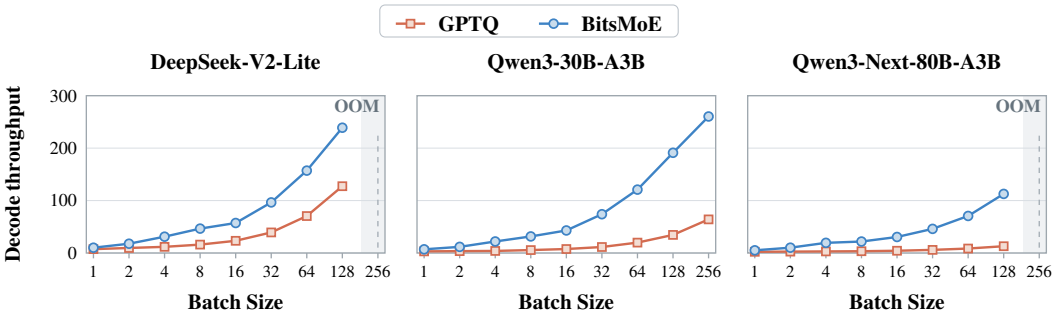


Figure 7: Decode throughput (tok/s) of GPTQ and BITSMOE at 2-bit on DeepSeek-V2-Lite, Qwen3-30B-A3B and Qwen3-Next-80B-A3B.

267.04 minutes, a $4.79\times$ speedup (Fig. 6a). On Qwen3-30B-A3B at 2-bit, it achieves $49.00\times$ and $16.47\times$ end-to-end offline quantization speedups over the ILP-based methods MxMoE and GEMQ, respectively. FQCM computes component costs directly for bit allocation, avoiding repeated empirical evaluation of candidate output errors. App. F.2 reports per-stage timings and the ILP-based comparison (Tab. 16).

Inference Efficiency. BITSMOE also reduces the GPU memory footprint of Qwen3-Next-80B-A3B from 148.69 GB in FP16 to 25.46 GB at 2-bit, a $5.84\times$ reduction (Fig. 6b). For decoding, we use the Triton backend (Tillet et al., 2019) for GPTQ because the Marlin (Frantar et al., 2025) and ExLlamaV2 (turboderp-org, 2023) backends in our GPTQModel implementation do not support 2-bit GPTQ checkpoints. On a single NVIDIA A6000 GPU, BITSMOE achieves higher decode throughput than GPTQ on all three models across all measured batch sizes (Fig. 7). Decode speedups over GPTQ reach up to $2.90\times$, $6.46\times$, and $8.71\times$ on DeepSeek-V2-Lite, Qwen3-30B-A3B, and Qwen3-Next-80B-A3B, respectively. The absolute throughput gap widens as batch size increases. App. F.1 provides further results and analysis on inference efficiency.

5 CONCLUSION

We presented BITSMOE, a fine-grained MoE quantization framework that defines structurally meaningful quantization units and models their costs under actual activation patterns and different bit-widths. SSD separates shared structure across experts from expert-specific spectral components, which serve as quantization units. FQCM combines intrinsic spectral importance, activation-dependent importance, and bit-width-dependent distortion to estimate component-wise costs for global ILP-based bit allocation under a fixed memory budget. Experiments on three MoE models and seven downstream tasks demonstrate improved average accuracy at 2-bit. On Qwen3-30B-A3B at 2-bit, BITSMOE achieves 64.29% average accuracy, exceeding GEMQ by 2.80 percentage points. Under the same setting, it achieves a $16.47\times$ end-to-end offline quantization speedup over GEMQ. It also achieves up to $6.46\times$ the decode throughput of GPTQ. At 1.4-bit, it achieves 45.07% average accuracy, surpassing 2-bit GPTQ's 42.65% by 2.42 percentage points.

AI USE STATEMENT

We used generative AI tools to assist with writing proofs and implementing our methods through experimental code. We designed the experimental framework and settings; these tools assisted with code implementation and debugging within that framework. We also used them to polish author-written English drafts for clarity and academic style, assist with figure preparation, and convert existing experimental results into LaTeX tables.

We did not use generative AI to propose or refine research hypotheses, develop theoretical models or conceptual frameworks, formulate mathematical claims, or provide key ideas for proofs. Nor did we use it to design or provide feedback on research methodology or experiments, generate synthetic datasets, clean or reformat datasets, conduct qualitative or thematic data analysis, or interpret results. We did not use it for literature searches or summaries.

We reviewed the revised text, checked the mathematical derivations, reviewed the AI-assisted code and ran the experiments, and checked figures and AI-generated LaTeX table entries against the original results. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

Sec. 3 describes the method and GPU execution scheme, and App. B provides the quantization and inference algorithms. App. C details the theoretical assumptions, derivations, and ILP budget formulation, while App. D.1 describes the calibration of the shared low-bit distortion coefficients. Sec. 4 and App. A document the model checkpoints, calibration and quantization settings, and evaluation protocols. App. F.1 specifies the hardware and benchmark settings for inference efficiency measurements. We will provide anonymized core code with the submission. Upon acceptance, we plan to publicly release the complete implementation, including quantization code, GPU kernels, evaluation and benchmarking scripts, environment dependencies, and run configurations, together with pre-quantized model weights.

REFERENCES

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pp. 2357–2367, 2019.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=dfqsW38v1X.

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

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A survey on mixture of experts in large language models. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Patrick CHen, Hsiang-Fu Yu, Inderjit S Dhillon, and Cho-Jui Hsieh. DRONE: Data-aware low-rank compression for large NLP models. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=sthiz9zeXGG.

Zhixuan Chen, Xing Hu, Dawei Yang, Zukang Xu, XUCHEN, Zhihang Yuan, Sifan Zhou, and JiangyongYu. MoEQuant: Enhancing quantization for mixture-of-experts large language models via expert-balanced sampling and affinity guidance. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=0epuNvt5Dj.

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

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

DeepSeek-AI. Deepseek-v3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

Jianing Deng, Song Wang, Dongwei Wang, Zijie Liu, Tianlong Chen, Huanrui Yang, and Jingtong Hu. GEMQ: Global expert-level mixed-precision quantization for moe LLMs. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=5Q4ts0pr9T.

Haojie Duanmu, Xiuhong Li, Zhihang Yuan, Size Zheng, Jiangfei Duan, Xingcheng Zhang, and Dahua Lin. MxMoE: Mixed-precision quantization for moe with accuracy and performance co-design. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=pXoZLGMNDm.

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

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

Elias Frantar, Roberto L Castro, Jiale Chen, Torsten Hoefler, and Dan Alistarh. Marlin: Mixed-precision auto-regressive parallel inference on large language models. In *Proceedings of the 30th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming*, pp. 239–251, 2025.

Hao Gu, Wei Li, Lujun Li, Zhu Qiyuan, Mark G. Lee, Shengjie Sun, Wei Xue, and Yike Guo. Delta decompression for moe-based LLMs compression. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ziezViPoN1.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024. URL https://www.gurobi.com.

Shwai He, Liang Ding, Daize Dong, Boan Liu, Fuqiang Yu, and Dacheng Tao. Pad-net: An efficient framework for dynamic networks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14354–14366, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021b. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Yen-Chang Hsu, Ting Hua, Sungen Chang, Qian Lou, Yilin Shen, and Hongxia Jin. Language model compression with weighted low-rank factorization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uPv9Y3gmAI5.

Beichen Huang, Yueming Yuan, ZELEI SHAO, and Minjia Zhang. Milo: Efficient quantized moe inference with mixture of low-rank compensators. In *Eighth Conference on Machine Learning and Systems*, 2025. URL https://openreview.net/forum?id=NXVXiJmhe1.

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, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Wei Li, Lujun Li, Hao Gu, You-Liang Huang, Mark G. Lee, Shengjie Sun, Wei Xue, and Yike Guo. Moe-SVD: Structured mixture-of-experts LLMs compression via singular value decomposition. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=acJ3vdFljk.

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

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

Enshu Liu, Junyi Zhu, Zinan Lin, Xuefei Ning, Matthew B. Blaschko, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. Efficient expert pruning for sparse mixture-of-experts language models: Enhancing performance and reducing inference costs, 2025. URL https://openreview.net/forum?id=TTUtPIpaol.

Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models. *arXiv preprint arXiv:2402.14800*, 2024.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Evan Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. OLMoe: Open mixture-of-experts language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xXTkbTBmqq.

Bowen Ping, Shuo Wang, Hanqing Wang, Xu Han, Yuzhuang Xu, Yukun Yan, Yun Chen, Baobao Chang, Zhiyuan Liu, and Maosong Sun. Delta-CoMe: Training-free delta-compression with mixed-precision for large language models. *Advances in Neural Information Processing Systems*, 37:31056–31077, 2024.

Qwen Team. Qwen3-Next-80B-A3B-Instruct. Hugging Face model card, 2025. URL https://huggingface.co/Qwen/Qwen3-Next-80B-A3B-Instruct.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

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

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. In *ICLR*, 2024. URL https://openreview.net/forum?id=8Wuvhh0LYW.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Lintang Sutawika, Hailey Schoelkopf, Leo Gao, Baber Abbasi, Stella Biderman, Jonathan Tow, ben fattori, Charles Lovering, farzanehnakhaee70, Jason Phang, Anish Thite, Fazz, Aflah, Niklas, Thomas Wang, sdtblck, nopperl, gakada, tttyuntian, researcher2, Julen Etxaniz, Chris, Hanwool Albert Lee, Leonid Sinev, Zdeněk Kasner, Kiersten Stokes, Khalid, KonradSzafer, Jeffrey Hsu, and Anjor Kanekar. Eleutherai/lm-evaluation-harness: v0.4.9.1, August 2025. URL https://doi.org/10.5281/zenodo.16737642.

Philippe Tillet, Hsiang-Tsung Kung, and David Cox. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pp. 10–19, 2019.

turboderp-org. ExLlamaV2. https://github.com/turboderp-org/exllamav2, 2023. GitHub repository. Accessed: 2026-05-05.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LNYIUouhdt.

Guangxuan Xiao, Ji Lin, Mickaël Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *ICML*, pp. 38087–38099, 2023. URL https://proceedings.mlr.press/v202/xiao23c.html.

Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. Openmoe: An early effort on open mixture-of-experts language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=1YDeZU8Lt5.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Yuanlin Duan, Wenqi Jia, Miao Yin, Yu Cheng, and Bo Yuan. Moe-i-squared: Compressing mixture of experts models through inter-expert pruning and intra-expert low-rank decomposition. *arXiv preprint arXiv:2411.01016*, 2024.

Zhihang Yuan, Yuzhang Shang, Yue Song, Dawei Yang, Qiang Wu, Yan Yan, and Guangyu Sun. ASVD: Activation-aware singular value decomposition for compressing large language models, 2025. URL https://openreview.net/forum?id=HyPofygOCT.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Appendix Contents

Organization. In this appendix, we provide further details as follows:

Notation **15**

A Details on Models and Evaluation **16**

B Algorithm Details. **16**

C Mathematical Formulation **18**

C.1 Shared-basis Spectral Decomposition (SSD) 18

C.2 Factorized Quantization Cost Modeling (FQCM) 20

C.3 Derivation of Bit-width-dependent Distortion 23

C.4 Global Bit Budget by Projection Type 26

C.5 Component-wise Loss and ILP Formulation 27

D Empirical Validation of Shared Distortion Coefficients **28**

D.1 Calibration of Low-bit Distortion Coefficients 28

D.2 Analysis of Component-wise Quantization Costs 28

D.3 Dispersion of Bit-width-dependent Distortion 29

E Additional Accuracy Results and Sensitivity Analysis **30**

E.1 Accuracy under Ultra-Low-Bit Quantization 30

E.2 Supplementary Ablation Results 31

E.3 Sensitivity to Calibration Data Composition 32

F Inference Efficiency and Offline Quantization Costs **32**

F.1 Inference Efficiency and GPU Memory Usage 32

F.2 Offline Quantization Time Breakdown 34

NOTATION

The notation table (Tab. 5) collects the core quantities used in App. C. We use $\ell \in [L]$ for MoE layers, $e \in [E]$ for experts, and $k \in [n^{(h)}]$ for spectral components.

Table 5: Core notation for App. C.

Symbols	Description
Projection Types and Dimensions	
$h \in \mathcal{H}$, $\mathcal{H}_{\text{in}}$, h_{dn}	Projection type in $\mathcal{H} = \{\texttt{gate_proj}, \texttt{up_proj}, \texttt{down_proj}\}$; $\mathcal{H}_{\text{in}}$ contains gate/up, and h_{dn} denotes down.
$m, n, n^{(h)}$	Spectral-direction dimension $m = d_{\text{out}}$ and spectral-component count $n = n^{(h)} = d_{\text{in}}$ in the full-component setting; $d_{\text{in}}, d_{\text{out}}$ are `gate_proj`/`up_proj` input/output dimensions.
Shared-basis Spectral Decomposition	
$\boldsymbol{\Phi}^{(h)}$, $\boldsymbol{\phi}_k^{(h)}$	Shared spectral basis across experts within a layer, and its k-th column.
$P_e^{(h)}$, $p_{e,k}^{(h)}$	Normalized expert-specific spectral matrix and its k-th column, a unit-norm expert-specific spectral direction.
$A_e^{(h)}$, $\boldsymbol{\alpha}_{e,k}^{(h)}$	Diagonal magnitude matrix and component magnitude.
$C_{e,k}^{(h)}$, $\left(\boldsymbol{\alpha}_{e,k}^{(h)}\right)^2$	Rank-one spectral component and its spectral energy (intrinsic spectral importance).
Factorized Quantization Cost Modeling	
$\widehat{\boldsymbol{\Phi}}^{(h)}$, $\widehat{\boldsymbol{\phi}}_k^{(h)}$	Shared spectral basis quantized to 8-bit precision using GPTQ, and its k-th column.
$W_{e,q\phi}^{(h)}$	Reference weight with only the shared spectral basis quantized.
$H_{g,e}^{(h)}$	Routing-weighted Hessian.
$\boldsymbol{\beta}_{e,k}^{(h)}$	Activation-dependent importance.
$\boldsymbol{\varepsilon}_{e,k,b}^{(h)}$	Quantization error vector of the expert-specific spectral direction.
$\mathcal{E}_{e,k,b}^{(h)}$	Bit-width-dependent distortion $\mathbb{E}_\epsilon \|\boldsymbol{\varepsilon}_{e,k,b}^{(h)}\|_2^2$.
κ_b	Shared low-bit coefficient for $b \in \{2, 3, 4\}$.
$\mathcal{L}_{e,P}^{(h)}(b)$	Expected routing-weighted output reconstruction loss from quantizing spectral directions with the quantized shared basis fixed.
$\mathcal{L}_{e,k}^{(h)}(b)$	Component-wise quantization cost of assigning bit-width b.
Global Bit Allocation	
$b \in \mathcal{B}$	Candidate bit-width in $\mathcal{B} = \{16, 8, 6, 4, 3, 2, 1, 0\}$.
$y_{\ell,e,k,b}^{(h)}$	Binary variable assigning bit-width b to component (ℓ, e, h, k); y collects all assignment variables.
$\mathcal{L}_{\text{FQCM}}^{(h)}(y)$	Global FQCM loss summed over all layers, experts, and spectral components for projection type h.
$\mathfrak{b}_{\text{eq}}$, $B_{\text{alloc}}^{(h)}(\mathfrak{b}_{\text{eq}})$	Target bit-width and per-type global normalized bit budget.

A Details on Models and Evaluation

Table 6: Configurations of the evaluated MoE LLMs. Parameter counts are in billions. The expert configuration is expressed as (routed experts + shared experts) × MoE layers; Top-K counts the routed experts selected per token. DeepSeek-V2-Lite's first dense layer is excluded from the MoE layer count. Qwen3-30B-A3B has no shared experts.

Model	Total Params.	Active Params.	Variant	#Experts	Top-K
DeepSeek-V2-Lite	15.7	2.4	Base	$(64+2)\times 26$	6
Qwen3-30B-A3B	30.5	3.3	Base	128×48	8
Qwen3-Next-80B-A3B	80	3	Instruct	$(512+1)\times 48$	10

We evaluate BitsMoE on three MoE LLMs spanning different parameter scales, training stages, and architectures: DeepSeek-V2-Lite (Liu et al., 2024), Qwen3-30B-A3B-Base (Yang et al., 2025), and Qwen3-Next-80B-A3B-Instruct (Qwen Team, 2025). Tab. 6 summarizes their parameter counts and expert configurations. The first two checkpoints are pretrained base models, whereas Qwen3-Next-80B-A3B is instruction-tuned. DeepSeek-V2-Lite combines Multi-head Latent Attention (MLA) with routed and shared experts; Qwen3-30B-A3B uses grouped-query attention with routed experts only; and Qwen3-Next-80B-A3B combines Gated DeltaNet and Gated Attention with a high-sparsity MoE architecture containing both routed and shared experts.

We assess both downstream task accuracy and language-modeling perplexity. The six zero-shot tasks are ARC-Easy (Clark et al., 2018), HellaSwag (Zellers et al., 2019), MathQA (Amini et al., 2019), MMLU (Hendrycks et al., 2021a), OpenBookQA (Mihaylov et al., 2018), and WinoGrande (Sakaguchi et al., 2021), covering science question answering, commonsense reasoning, mathematical reasoning, and broad knowledge. We additionally evaluate multi-step mathematical reasoning on GSM8K (Cobbe et al., 2021). All seven downstream tasks are evaluated using lm-evaluation-harness (version 0.4.9.1) (Sutawika et al., 2025). We also report WikiText-2 (Merity et al., 2016) perplexity for Qwen3-30B-A3B.

Accuracy and perplexity are evaluated on NVIDIA A100 GPUs. Average accuracy is computed over the seven downstream tasks. GSM8K is evaluated in a 5-shot setting. Chat templates are applied to instruction-tuned models. WikiText-2 perplexity is evaluated on the test split of `wikitext-2-raw-v1` using the strided sliding-window strategy described in the HuggingFace Transformers documentation, with a window length of 2,048 tokens and a stride of 512 tokens. For calibration, we use 1,024 samples from the English C4 training split (Raffel et al., 2020), with a maximum sequence length of 2,048 tokens and sampling seed 42. The C4–MATH calibration mixtures are examined separately in App. E.3. Shared bases are quantized to 8-bit precision using GPTQ; magnitude matrices $A_e^{(h)}$ and stored scales use FP16. Routers, shared experts, and non-expert modules are not quantized.

B Algorithm Details

Algorithm 1 collects costs over all L MoE layers before solving three independent global ILPs, one for each projection type. Following the notation table (Tab. 5), we use $m = d_{\text{out}}$ and $n = n^{(h)} = d_{\text{in}}$ in the full-component setting. Each type receives one third of the global bit budget B remaining after reserving the 8-bit shared bases, giving the global normalized bit budget $B_{\text{alloc}}^{(h)} = \lfloor B/(3m) \rfloor$ derived in App. C.4. Calibration statistics are collected with the original FP16 model.

The shared spectral basis is quantized to 8-bit precision using GPTQ, yielding $\widehat{\mathbf{\Phi}}_\ell^{(h)}$. The expert-specific spectral directions and component magnitudes are retained from SSD for subsequent bit allocation and quantization.

Algorithm 1 BITSMOE Offline Quantization

Require: Pretrained model $\mathcal{M}$, calibration data $\mathcal{D}$, target bit-width $\mathfrak{b}_{\text{eq}}$

Ensure: Quantized model with packed expert directions and 8-bit GPTQ shared bases

$\mathcal{B} \leftarrow \{16, 8, 6, 4, 3, 2, 1, 0\}$
for each MoE layer $\ell \in [L]$ in $\mathcal{M}$ **do**
  Collect original projection inputs, routing weights, and unweighted and routing-weighted second moments on $\mathcal{D}$.
  for each projection type $h \in \mathcal{H}$ **do**
    Concatenate $\{W_{\ell,e}^{(h)}\}_{e=1}^{E}$ vertically for `gate_proj` and `up_proj`, and horizontally for `down_proj`.
    Apply SSD to obtain $\mathbf{\Phi}_{\ell}^{(h)}$ and expert factors $\{\widetilde{P}_{\ell,e}^{(h)}\}_{e=1}^{E}$.
    Normalize the expert factors into directions $P_{\ell,e}^{(h)}$ and magnitude matrices $A_{\ell,e}^{(h)}$, for all e.
    GPTQ-quantize the shared transform to 8-bit precision using its unweighted input statistics, yielding $\widehat{\mathbf{\Phi}}_{\ell}^{(h)}$.
    Compute activation-dependent importance from Eq. (39), with the diagonal approximation and normalization described in App. C.2.
    Build and cache the component-wise quantization costs $\{\mathcal{L}_{\ell,e,k}^{(h)}(b)\}_{e,k,b}$ from squared magnitudes, activation-dependent importance, and bit-width-dependent distortion.
    Cache $P_{\ell,e}^{(h)}$, $A_{\ell,e}^{(h)}$, and the GPTQ shared-basis integers and scales.
  end for
end for
$B \leftarrow 3LEmn\mathfrak{b}_{\text{eq}} - 8 \cdot 3Ln^2$
for each projection type $h \in \mathcal{H}$ **do**
  $B_{\text{alloc}}^{(h)}(\mathfrak{b}_{\text{eq}}) \leftarrow \lfloor B/(3m) \rfloor$
  Minimize the global FQCM loss in Eq. (9) through the ILP in Eq. (10) for this type, obtaining $\{y_{\ell,e,k,b}^{(h)}\}_{\ell,e,k,b}$.
end for
for each layer ℓ, expert e, and projection type h **do**
  $b_{\ell,e,k}^{(h)} \leftarrow \sum_{b \in \mathcal{B}} b\, y_{\ell,e,k,b}^{(h)}, \quad \text{for all } k \in [n]$
  Quantize each direction as $Q_{b_{\ell,e,k}^{(h)}}(p_{\ell,e,k}^{(h)})$ using group size 128.
  Omit $b = 0$ and retain $b = 16$ in FP16.
  Group directions by bit-width; bit-plane pack groups with $1 \le b \le 8$ and store their original indices and scales.
  Retain $A_{\ell,e}^{(h)}$ in FP16.
end for
Pack the 8-bit shared-basis integers and FP16 scales for Marlin W8A16 execution, once per layer and projection type.
return Quantized model

Algorithm 2 describes the forward pass for a single token in an MoE layer. Matrix–vector multiplication with $\widehat{P}_e^{(h)}$ or its transpose uses stored component indices and on-the-fly tile dequantization of the packed spectral directions. Matrix–vector multiplication with the shared bases uses $\mathrm{Marlin}_{\mathrm{W8A16}}(W, a)$, which evaluates Wa with W stored as packed 8-bit integers with FP16 scales and a cast to FP16. Each shared transform is applied once per token: `gate_proj` and `up_proj` before expert computation, and `down_proj` after accumulating the routing-weighted expert contributions.

Algorithm 2 BITSMOE Forward Pass with Packed Expert Directions

Require: Token input $x \in \mathbb{R}^n$, packed directions $\{\widehat{P}_e^{(h)}\}$, FP16 magnitudes $\{A_e^{(h)}\}$, 8-bit shared bases $\{\widehat{\mathbf{\Phi}}^{(h)}\}$ in Marlin format

Ensure: MoE output $y \in \mathbb{R}^n$

Obtain routed experts $\mathcal{T}(x)$ and weights $\{g_e(x)\}$ from the router.
for each $h \in \mathcal{H}_{\mathrm{in}}$ **do**
  $z^{(h)} \leftarrow \mathrm{Marlin}_{\mathrm{W8A16}}\Big(\big(\widehat{\mathbf{\Phi}}^{(h)}\big)^\top, x\Big)$
end for
$v \leftarrow 0 \in \mathbb{R}^n$
for each $e \in \mathcal{T}(x)$ **do**
  for each $h \in \mathcal{H}_{\mathrm{in}}$ **do**
    $u_e^{(h)} \leftarrow \widehat{P}_e^{(h)} A_e^{(h)} z^{(h)}$
  end for
  $a_e \leftarrow \mathrm{SiLU}\big(u_e^{(\texttt{gate_proj})}\big) \odot u_e^{(\texttt{up_proj})}$
  $v \leftarrow v + g_e(x) A_e^{(h_{\mathrm{dn}})} \big(\widehat{P}_e^{(h_{\mathrm{dn}})}\big)^\top a_e$
end for
$y \leftarrow \mathrm{Marlin}_{\mathrm{W8A16}}\Big(\widehat{\mathbf{\Phi}}^{(h_{\mathrm{dn}})}, v\Big)$
Add the original shared-expert branch, including its gate, if present.
return y

C MATHEMATICAL FORMULATION

This section develops the mathematical formulation underlying Shared-basis Spectral Decomposition (SSD) and Factorized Quantization Cost Modeling (FQCM) in BITSMOE. We first formalize SSD for all three expert projection types, separating a shared spectral basis, component magnitudes, and expert-specific unit-norm directions that serve as quantization units. Building on this representation, we derive the output reconstruction loss induced by quantizing the expert-specific spectral directions and state the approximations through which FQCM combines intrinsic spectral importance, activation-dependent importance, and bit-width-dependent distortion into component-wise costs. Finally, we specify the piecewise distortion model, translate the target bit-width into an allocation budget, and formulate the resulting component-wise ILP problem. Following the notation table (Tab. 5), $\ell \in [L]$ indexes MoE layers, $e \in [E]$ indexes experts, $k \in [n^{(h)}]$ indexes spectral components, $b \in \mathcal{B}$ denotes a candidate bit-width, and $h \in \mathcal{H}$ indexes projection types. Thus, fully indexed quantities take forms such as $p_{\ell,e,k}^{(h)}$ and $\mathcal{E}_{\ell,e,k,b}^{(h)}$. For concise presentation, we omit the layer index ℓ in App. C.1 and App. C.2, and omit ℓ, e, and h in App. C.3.

C.1 SHARED-BASIS SPECTRAL DECOMPOSITION (SSD)

The SSD construction applies to every MoE layer; throughout App. C.1, we fix an arbitrary layer and omit the layer index ℓ.

Within an MoE layer, expert weights operate on the same feature space and often exhibit substantial redundancy. Instead of decomposing each expert independently, we jointly decompose all experts to extract a shared spectral basis. We denote the expert projection types by

$\mathcal{H} := \{\texttt{gate_proj}, \texttt{up_proj}, \texttt{down_proj}\}$. We group `gate_proj` and `up_proj` into $\mathcal{H}_{\text{in}} := \{\texttt{gate_proj}, \texttt{up_proj}\}$, for which the shared bases act on the input feature space, and denote `down_proj` by h_{dn}, for which the shared basis acts on the output feature space. Following Eq. (1), d_{in} and d_{out} denote the input and output dimensions of the `gate_proj` and `up_proj` weights. Thus, $W_e^{(h)} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ for $h \in \mathcal{H}_{\text{in}}$, whereas $W_e^{(h_{\text{dn}})} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$.

For $h \in \mathcal{H}_{\text{in}}$, expert weights share the same input feature space. We concatenate expert weights along the output-channel dimension:

$$W_{\text{cat}}^{(h)} := \begin{bmatrix} W_1^{(h)} \\ \vdots \\ W_E^{(h)} \end{bmatrix}. \tag{11}$$

We then compute the SVD of the concatenated matrix and merge the singular values into the left factor:

$$W_{\text{cat}}^{(h)} = U_{\text{cat}}^{(h)} \mathbf{\Sigma}^{(h)} \big(\mathbf{\Phi}^{(h)}\big)^\top = \widetilde{P}_{\text{cat}}^{(h)} \big(\mathbf{\Phi}^{(h)}\big)^\top, \qquad \widetilde{P}_{\text{cat}}^{(h)} := U_{\text{cat}}^{(h)} \mathbf{\Sigma}^{(h)}. \tag{12}$$

After merging the singular values, we partition the resulting matrix according to expert blocks:

$$\widetilde{P}_{\text{cat}}^{(h)} = \begin{bmatrix} \widetilde{P}_1^{(h)} \\ \vdots \\ \widetilde{P}_E^{(h)} \end{bmatrix}, \qquad W_e^{(h)} = \widetilde{P}_e^{(h)} \big(\mathbf{\Phi}^{(h)}\big)^\top. \tag{13}$$

For $h = h_{\text{dn}}$, expert weights share the same output feature space. We concatenate expert weights along the input-channel dimension:

$$W_{\text{cat}}^{(h)} := \begin{bmatrix} W_1^{(h)} & \cdots & W_E^{(h)} \end{bmatrix}. \tag{14}$$

The corresponding SVD-and-absorption step is

$$W_{\text{cat}}^{(h)} = \mathbf{\Phi}^{(h)} \mathbf{\Sigma}^{(h)} \big(V_{\text{cat}}^{(h)}\big)^\top = \mathbf{\Phi}^{(h)} \big(\widetilde{P}_{\text{cat}}^{(h)}\big)^\top, \qquad \widetilde{P}_{\text{cat}}^{(h)} := V_{\text{cat}}^{(h)} \mathbf{\Sigma}^{(h)}. \tag{15}$$

Partitioning $\widetilde{P}_{\text{cat}}^{(h)}$ according to expert input-channel blocks gives

$$\widetilde{P}_{\text{cat}}^{(h)} = \begin{bmatrix} \widetilde{P}_1^{(h)} \\ \vdots \\ \widetilde{P}_E^{(h)} \end{bmatrix}, \qquad W_e^{(h)} = \mathbf{\Phi}^{(h)} \big(\widetilde{P}_e^{(h)}\big)^\top. \tag{16}$$

Definition C.1 (Spectral component and magnitude matrix)**.** Let $\boldsymbol{\phi}_k^{(h)}$ denote the k-th column of the shared basis $\mathbf{\Phi}^{(h)}$, and let $\widetilde{p}_{e,k}^{(h)} := \widetilde{P}_e^{(h)}[:, k]$ denote the corresponding expert-specific spectral vector. We define its magnitude as

$$\boldsymbol{\alpha}_{e,k}^{(h)} := \left\| \widetilde{p}_{e,k}^{(h)} \right\|_2. \tag{17}$$

The component magnitudes are collected into a diagonal matrix

$$A_e^{(h)} := \text{Diag}(\boldsymbol{\alpha}_{e,1}^{(h)}, \ldots, \boldsymbol{\alpha}_{e,n^{(h)}}^{(h)}). \tag{18}$$

The corresponding rank-one spectral component is

$$C_{e,k}^{(h)} := \begin{cases} \widetilde{p}_{e,k}^{(h)} \big(\boldsymbol{\phi}_k^{(h)}\big)^\top, & h \in \mathcal{H}_{\text{in}}, \\ \boldsymbol{\phi}_k^{(h)} \big(\widetilde{p}_{e,k}^{(h)}\big)^\top, & h = h_{\text{dn}}. \end{cases} \tag{19}$$

Definition C.2 (Normalized expert-specific spectral matrix)**.** To decouple spectral component magnitude from direction, each column of $\widetilde{P}_e^{(h)}$ is normalized by its magnitude:

$$P_e^{(h)} := [p_{e,1}^{(h)}, \ldots, p_{e,n^{(h)}}^{(h)}], \qquad p_{e,k}^{(h)} := \frac{\widetilde{p}_{e,k}^{(h)}}{\boldsymbol{\alpha}_{e,k}^{(h)}}. \tag{20}$$

Each expert-specific spectral direction satisfies $p_{e,k}^{(h)} \in \mathbb{R}^m$, where $m = d_{\text{out}}$ denotes its dimension for all three projection types. Thus, $\|p_{e,k}^{(h)}\|_2 = 1$ for every spectral component. The expert weight admits the unified normalized shared-basis form

$$W_e^{(h)} = \begin{cases} P_e^{(h)} A_e^{(h)} \big(\boldsymbol{\Phi}^{(h)}\big)^\top = \sum_{k=1}^{n^{(h)}} \boldsymbol{\alpha}_{e,k}^{(h)} p_{e,k}^{(h)} \big(\boldsymbol{\phi}_k^{(h)}\big)^\top, & h \in \mathcal{H}_{\text{in}}, \\ \boldsymbol{\Phi}^{(h)} A_e^{(h)} \big(P_e^{(h)}\big)^\top = \sum_{k=1}^{n^{(h)}} \boldsymbol{\alpha}_{e,k}^{(h)} \boldsymbol{\phi}_k^{(h)} \big(p_{e,k}^{(h)}\big)^\top, & h = h_{\text{dn}}. \end{cases} \tag{21}$$

For each projection type $h \in \mathcal{H} = \{\texttt{gate_proj}, \texttt{up_proj}, \texttt{down_proj}\}$, the number of spectral components is $n^{(h)} = \min(Ed_{\text{out}}, d_{\text{in}})$, with $\boldsymbol{\Phi}^{(h)} \in \mathbb{R}^{d_{\text{in}} \times n^{(h)}}$ and $P_e^{(h)} \in \mathbb{R}^{d_{\text{out}} \times n^{(h)}}$.

Spectral energy. Following Eq. (3), we define the spectral energy of spectral component $C_{e,k}^{(h)}$ as its squared Frobenius norm:

$$\begin{aligned} \left\| C_{e,k}^{(h)} \right\|_F^2 &= \operatorname{Tr}\Big[\big(C_{e,k}^{(h)}\big)^\top C_{e,k}^{(h)} \Big] \\ &= \big(\boldsymbol{\alpha}_{e,k}^{(h)}\big)^2 \left\| p_{e,k}^{(h)} \right\|_2^2 \left\| \boldsymbol{\phi}_k^{(h)} \right\|_2^2 = \big(\boldsymbol{\alpha}_{e,k}^{(h)}\big)^2. \end{aligned} \tag{22}$$

The last equality follows from the unit norms of the expert-specific direction $p_{e,k}^{(h)}$ and the shared basis direction $\boldsymbol{\phi}_k^{(h)}$. This identity holds for every $h \in \mathcal{H}$.

Each spectral component combines a shared basis direction $\boldsymbol{\phi}_k^{(h)}$ with an expert-specific magnitude $\boldsymbol{\alpha}_{e,k}^{(h)}$ and normalized spectral direction $p_{e,k}^{(h)}$. We therefore use the expert-specific spectral direction $p_{e,k}^{(h)}$ of each component as the quantization unit in BITSMOE, assigning a bit-width to each direction.

C.2 FACTORIZED QUANTIZATION COST MODELING (FQCM)

The FQCM derivation applies to every MoE layer; throughout App. C.2, we fix an arbitrary layer and omit the layer index ℓ.

For each projection type $h \in \mathcal{H}$, we first quantize the shared basis to 8-bit precision using GPTQ (Frantar et al., 2022) and denote the resulting basis by $\widehat{\boldsymbol{\Phi}}^{(h)}$.

For expert $e \in [E]$ and projection type $h \in \mathcal{H}_{\text{in}}$, we apply b-bit quantization to the expert-specific spectral directions obtained by SSD:

$$\widehat{p}_{e,k}^{(h)} = Q_b(p_{e,k}^{(h)}), \qquad \boldsymbol{\varepsilon}_{e,k,b}^{(h)} := p_{e,k}^{(h)} - \widehat{p}_{e,k}^{(h)}. \tag{23}$$

Using the quantized shared basis $\widehat{\boldsymbol{\Phi}}^{(h)} = [\widehat{\boldsymbol{\phi}}_1^{(h)}, \ldots, \widehat{\boldsymbol{\phi}}_{n^{(h)}}^{(h)}]$, quantizing the expert-specific spectral directions gives the reconstructed weight

$$\widehat{W}_e^{(h)} = \sum_{k=1}^{n^{(h)}} \boldsymbol{\alpha}_{e,k}^{(h)} \widehat{p}_{e,k}^{(h)} \big(\widehat{\boldsymbol{\phi}}_k^{(h)}\big)^\top. \tag{24}$$

Let $W_{e,q\phi}^{(h)} := P_e^{(h)} A_e^{(h)} \big(\widehat{\boldsymbol{\Phi}}^{(h)}\big)^\top$. We measure the additional error introduced by quantizing the expert-specific spectral directions relative to $W_{e,q\phi}^{(h)}$ and define the corresponding weight perturbation as

$$\boldsymbol{\Delta}_e^{(h)} := W_{e,q\phi}^{(h)} - \widehat{W}_e^{(h)} = \sum_{k=1}^{n^{(h)}} \boldsymbol{\alpha}_{e,k}^{(h)} \boldsymbol{\varepsilon}_{e,k,b}^{(h)} \big(\widehat{\boldsymbol{\phi}}_k^{(h)}\big)^\top. \tag{25}$$

We define the quantization error matrix of the expert-specific spectral directions as $\boldsymbol{E}_{P,e}^{(h)} := P_e^{(h)} - \widehat{P}_e^{(h)} = [\boldsymbol{\varepsilon}_{e,1,b}^{(h)}, \ldots, \boldsymbol{\varepsilon}_{e,n^{(h)},b}^{(h)}]$. The weight perturbation can then be written as $\boldsymbol{\Delta}_e^{(h)} = \boldsymbol{E}_{P,e}^{(h)} A_e^{(h)} \big(\widehat{\boldsymbol{\Phi}}^{(h)}\big)^\top$.

Definition C.3 (Routing-weighted Hessian)**.** For expert e and projection type $h \in \mathcal{H}_{\text{in}}$, let $X_e^{(h)} = [x_{e,1}^{(h)}, \ldots, x_{e,t}^{(h)}] \in \mathbb{R}^{d_{\text{in}} \times t}$ contain the t token activations routed to this expert, and let $g_e = [g_{e,1}, \ldots, g_{e,t}]^\top \in \mathbb{R}^t$ contain the corresponding routing weights. With $G_e := \mathrm{Diag}(g_e)$, we define the routing-weighted Hessian as

$$H_{g,e}^{(h)} := X_e^{(h)} G_e \big(X_e^{(h)}\big)^\top = \sum_{j=1}^{t} g_{e,j} x_{e,j}^{(h)} \big(x_{e,j}^{(h)}\big)^\top. \tag{26}$$

Since $g_{e,j} \geq 0$, $H_{g,e}^{(h)} \succeq 0$. Writing $X_{g,e}^{(h)} := X_e^{(h)} G_e^{1/2}$ gives $H_{g,e}^{(h)} = X_{g,e}^{(h)} \big(X_{g,e}^{(h)}\big)^\top$.

Under this metric, we define the expected routing-weighted output reconstruction loss from quantizing the spectral directions as

$$\mathcal{L}_{e,P}^{(h)}(b) := \mathbb{E}_\epsilon \left\| (W_{e,q\phi}^{(h)} - \widehat{W}_e^{(h)}) X_{g,e}^{(h)} \right\|_F^2, \tag{27}$$

where $\mathbb{E}_\epsilon$ denotes expectation over the quantization errors. The subscript P identifies the quantization of expert-specific spectral directions with the quantized shared basis fixed, as in Eq. (6); the dependence of $\widehat{W}_e^{(h)}$ and $\widehat{P}_e^{(h)}$ on b is implicit.

Factorized reconstruction-loss approximation. For expert e and projection type $h \in \mathcal{H}_{\text{in}}$, the expected routing-weighted output reconstruction loss admits the following additive approximation when cross-component quantization-error terms are neglected:

$$\mathcal{L}_{e,P}^{(h)}(b) \approx \sum_{k=1}^{n^{(h)}} \big(\boldsymbol{\alpha}_{e,k}^{(h)}\big)^2 \boldsymbol{\beta}_{e,k}^{(h)} \mathcal{E}_{e,k,b}^{(h)}, \tag{28}$$

where the activation-dependent importance and bit-width-dependent distortion are

$$\begin{aligned} \boldsymbol{\beta}_{e,k}^{(h)} &:= \big(\widehat{\boldsymbol{\phi}}_k^{(h)}\big)^\top H_{g,e}^{(h)} \widehat{\boldsymbol{\phi}}_k^{(h)}, \\ \mathcal{E}_{e,k,b}^{(h)} &:= \mathbb{E}_\epsilon \left\| \boldsymbol{\varepsilon}_{e,k,b}^{(h)} \right\|_2^2. \end{aligned} \tag{29}$$

Derivation. Starting from Eq. (27) and substituting the perturbation in Eq. (25), we obtain

$$\begin{aligned} \mathcal{L}_{e,P}^{(h)}(b) &= \mathbb{E}_\epsilon \left\| \big(W_{e,q\phi}^{(h)} - \widehat{W}_e^{(h)}\big) X_{g,e}^{(h)} \right\|_F^2 \\ &= \mathbb{E}_\epsilon \left\| \boldsymbol{\Delta}_e^{(h)} X_{g,e}^{(h)} \right\|_F^2 \\ &= \mathbb{E}_\epsilon \,\mathrm{Tr}\Big[\boldsymbol{\Delta}_e^{(h)} X_{g,e}^{(h)} \big(X_{g,e}^{(h)}\big)^\top \big(\boldsymbol{\Delta}_e^{(h)}\big)^\top \Big] \\ &= \mathbb{E}_\epsilon \,\mathrm{Tr}\Big[\boldsymbol{\Delta}_e^{(h)} H_{g,e}^{(h)} \big(\boldsymbol{\Delta}_e^{(h)}\big)^\top \Big] \\ &= \sum_{k=1}^{n^{(h)}} \sum_{l=1}^{n^{(h)}} \boldsymbol{\alpha}_{e,k}^{(h)} \boldsymbol{\alpha}_{e,l}^{(h)} \mathbb{E}_\epsilon \,\mathrm{Tr}\Big[\boldsymbol{\varepsilon}_{e,k,b}^{(h)} \big(\widehat{\boldsymbol{\phi}}_k^{(h)}\big)^\top H_{g,e}^{(h)} \widehat{\boldsymbol{\phi}}_l^{(h)} \big(\boldsymbol{\varepsilon}_{e,l,b}^{(h)}\big)^\top \Big] \\ &= \sum_{k=1}^{n^{(h)}} \sum_{l=1}^{n^{(h)}} \boldsymbol{\alpha}_{e,k}^{(h)} \boldsymbol{\alpha}_{e,l}^{(h)} \Big(\big(\widehat{\boldsymbol{\phi}}_k^{(h)}\big)^\top H_{g,e}^{(h)} \widehat{\boldsymbol{\phi}}_l^{(h)} \Big) \mathbb{E}_\epsilon \Big[\big(\boldsymbol{\varepsilon}_{e,k,b}^{(h)}\big)^\top \boldsymbol{\varepsilon}_{e,l,b}^{(h)} \Big]. \end{aligned} \tag{30}$$

For $b \geq 1$, we model the quantization errors as symmetric about zero, giving $\mathbb{E}_\epsilon[\boldsymbol{\varepsilon}_{e,k,b}^{(h)}] = 0$. Assuming that quantization errors across distinct spectral components are approximately uncorrelated, we obtain

$$\mathbb{E}_\epsilon \Big[\big(\boldsymbol{\varepsilon}_{e,k,b}^{(h)}\big)^\top \boldsymbol{\varepsilon}_{e,l,b}^{(h)} \Big] \approx \mathbb{E}_\epsilon \Big[\boldsymbol{\varepsilon}_{e,k,b}^{(h)} \Big]^\top \mathbb{E}_\epsilon \Big[\boldsymbol{\varepsilon}_{e,l,b}^{(h)} \Big] = 0, \qquad k \neq l. \tag{31}$$

For zero-bit eviction ($b = 0$), where $\boldsymbol{\varepsilon}_{e,k,0}^{(h)} = p_{e,k}^{(h)}$, we retain the same diagonal approximation in the allocation objective.

Separating the diagonal and cross-component terms in Eq. (30) gives

$$\begin{aligned}\mathcal{L}_{e,P}^{(h)}(b) &= \sum_{k=1}^{n^{(h)}} \big(\boldsymbol{\alpha}_{e,k}^{(h)}\big)^2 \boldsymbol{\beta}_{e,k}^{(h)} \mathcal{E}_{e,k,b}^{(h)} \\ &\quad + \sum_{\substack{k,l=1 \\ k\neq l}}^{n^{(h)}} \boldsymbol{\alpha}_{e,k}^{(h)} \boldsymbol{\alpha}_{e,l}^{(h)} \left(\big(\widehat{\boldsymbol{\phi}}_k^{(h)}\big)^\top H_{g,e}^{(h)} \widehat{\boldsymbol{\phi}}_l^{(h)} \right) \mathbb{E}_\epsilon \Big[\big(\boldsymbol{\varepsilon}_{e,k,b}^{(h)}\big)^\top \boldsymbol{\varepsilon}_{e,l,b}^{(h)} \Big] \\ &\approx \sum_{k=1}^{n^{(h)}} \big(\boldsymbol{\alpha}_{e,k}^{(h)}\big)^2 \boldsymbol{\beta}_{e,k}^{(h)} \mathcal{E}_{e,k,b}^{(h)}.\end{aligned} \tag{32}$$

Only the final step invokes the diagonal approximation, yielding Eq. (28) and the factorization in Eq. (7). Since $H_{g,e}^{(h)} \succeq 0$, $\boldsymbol{\beta}_{e,k}^{(h)} \geq 0$.

For $h = h_{\mathrm{dn}}$, the input activations satisfy $X_e^{(h)} \in \mathbb{R}^{d_{\mathrm{out}} \times t}$, and the quantized shared basis $\widehat{\boldsymbol{\Phi}}^{(h)}$ acts in the output space. Define $W_{e,q\phi}^{(h)} := \widehat{\boldsymbol{\Phi}}^{(h)} A_e^{(h)} \big(P_e^{(h)}\big)^\top$ and $\widehat{W}_e^{(h)} := \widehat{\boldsymbol{\Phi}}^{(h)} A_e^{(h)} \big(\widehat{P}_e^{(h)}\big)^\top$. The corresponding weight perturbation from quantizing the expert-specific spectral directions is

$$\begin{aligned}\boldsymbol{\Delta}_e^{(h)} &:= W_{e,q\phi}^{(h)} - \widehat{W}_e^{(h)} = \widehat{\boldsymbol{\Phi}}^{(h)} A_e^{(h)} \big(\boldsymbol{E}_{P,e}^{(h)}\big)^\top \\ &= \sum_{k=1}^{n^{(h)}} \boldsymbol{\alpha}_{e,k}^{(h)} \widehat{\boldsymbol{\phi}}_k^{(h)} \big(\boldsymbol{\varepsilon}_{e,k,b}^{(h)}\big)^\top.\end{aligned} \tag{33}$$

Using $H_{g,e}^{(h)} = X_e^{(h)} G_e \big(X_e^{(h)}\big)^\top \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{out}}}$, define the shared-basis Gram matrix $M^{(h)} := \big(\widehat{\boldsymbol{\Phi}}^{(h)}\big)^\top \widehat{\boldsymbol{\Phi}}^{(h)}$, with entries $M_{kl}^{(h)} = \big(\widehat{\boldsymbol{\phi}}_k^{(h)}\big)^\top \widehat{\boldsymbol{\phi}}_l^{(h)}$. The expected routing-weighted output reconstruction loss expands as

$$\begin{aligned}\mathcal{L}_{e,P}^{(h)}(b) &= \mathbb{E}_\epsilon \left\| \boldsymbol{\Delta}_e^{(h)} X_{g,e}^{(h)} \right\|_F^2 \\ &= \mathbb{E}_\epsilon \operatorname{Tr}\Big[\widehat{\boldsymbol{\Phi}}^{(h)} A_e^{(h)} \big(\boldsymbol{E}_{P,e}^{(h)}\big)^\top H_{g,e}^{(h)} \boldsymbol{E}_{P,e}^{(h)} A_e^{(h)} \big(\widehat{\boldsymbol{\Phi}}^{(h)}\big)^\top \Big] \\ &= \mathbb{E}_\epsilon \operatorname{Tr}\Big[M^{(h)} A_e^{(h)} \big(\boldsymbol{E}_{P,e}^{(h)}\big)^\top H_{g,e}^{(h)} \boldsymbol{E}_{P,e}^{(h)} A_e^{(h)} \Big] \\ &= \sum_{k=1}^{n^{(h)}} \sum_{l=1}^{n^{(h)}} \boldsymbol{\alpha}_{e,k}^{(h)} \boldsymbol{\alpha}_{e,l}^{(h)} M_{kl}^{(h)} \mathbb{E}_\epsilon \Big[\big(\boldsymbol{\varepsilon}_{e,k,b}^{(h)}\big)^\top H_{g,e}^{(h)} \boldsymbol{\varepsilon}_{e,l,b}^{(h)} \Big].\end{aligned} \tag{34}$$

We assume that the shared-basis Gram matrix satisfies

$$\left\| M^{(h)} - I \right\|_2 = \left\| \big(\widehat{\boldsymbol{\Phi}}^{(h)}\big)^\top \widehat{\boldsymbol{\Phi}}^{(h)} - I \right\|_2 \leq \delta_{\mathrm{orth}}^{(h)}, \qquad 0 \leq \delta_{\mathrm{orth}}^{(h)} \ll 1, \tag{35}$$

where $\|\cdot\|_2$ denotes the spectral norm and $\delta_{\mathrm{orth}}^{(h)}$ bounds the deviation from orthonormality.

Under this assumption, substituting $M^{(h)} \approx I$ into Eq. (34) yields

$$\mathcal{L}_{e,P}^{(h)}(b) \approx \sum_{k=1}^{n^{(h)}} \big(\boldsymbol{\alpha}_{e,k}^{(h)}\big)^2 \mathbb{E}_\epsilon \Big[\big(\boldsymbol{\varepsilon}_{e,k,b}^{(h)}\big)^\top H_{g,e}^{(h)} \boldsymbol{\varepsilon}_{e,k,b}^{(h)} \Big]. \tag{36}$$

In Eq. (36), the activation-weighted quantization error depends on the error direction and hence on the chosen bit-width. To decouple activation-dependent importance from quantization, we use the original expert-specific spectral direction $p_{e,k}^{(h)}$ obtained from SSD to measure the component's importance under the calibration activation patterns. For nonzero expected distortion, this modeling choice gives the surrogate

$$\frac{\mathbb{E}_\epsilon \Big[\big(\boldsymbol{\varepsilon}_{e,k,b}^{(h)}\big)^\top H_{g,e}^{(h)} \boldsymbol{\varepsilon}_{e,k,b}^{(h)} \Big]}{\mathbb{E}_\epsilon \left\| \boldsymbol{\varepsilon}_{e,k,b}^{(h)} \right\|_2^2} \approx \big(p_{e,k}^{(h)}\big)^\top H_{g,e}^{(h)} p_{e,k}^{(h)}. \tag{37}$$

This choice yields the component-wise cost approximation

$$\mathbb{E}_\epsilon\left[\left(\boldsymbol{\varepsilon}_{e,k,b}^{(h)}\right)^\top H_{g,e}^{(h)}\boldsymbol{\varepsilon}_{e,k,b}^{(h)}\right] \approx \boldsymbol{\beta}_{e,k}^{(h)}\mathbb{E}_\epsilon\left\|\boldsymbol{\varepsilon}_{e,k,b}^{(h)}\right\|_2^2, \qquad \boldsymbol{\beta}_{e,k}^{(h)} := \left(p_{e,k}^{(h)}\right)^\top H_{g,e}^{(h)} p_{e,k}^{(h)}. \tag{38}$$

For $b = 0$, this component-wise relation is exact because $\boldsymbol{\varepsilon}_{e,k,0}^{(h)} = p_{e,k}^{(h)}$ and $\|p_{e,k}^{(h)}\|_2 = 1$.

The activation-dependent importance used in the FQCM model is thus

$$\boldsymbol{\beta}_{e,k}^{(h)} := \begin{cases} \left(\widehat{\boldsymbol{\phi}}_k^{(h)}\right)^\top H_{g,e}^{(h)}\widehat{\boldsymbol{\phi}}_k^{(h)}, & h \in \mathcal{H}_{\mathrm{in}}, \\ \left(p_{e,k}^{(h)}\right)^\top H_{g,e}^{(h)} p_{e,k}^{(h)}, & h = h_{\mathrm{dn}}. \end{cases} \tag{39}$$

Activation-dependent importance for allocation. For bit allocation, we evaluate Eq. (39) using only the diagonal of the routing-weighted Hessian and normalize the result by the mean diagonal of the unweighted activation second moment $X_e^{(h)}\left(X_e^{(h)}\right)^\top$. This normalization removes differences in overall activation scale across layers, making the activation-dependent importance more comparable for global allocation while retaining expert-specific routing weights through G_e. The diagonal approximation retains routing-weighted per-coordinate second moments and omits cross-coordinate correlations, aiming to reduce sensitivity to noise in correlation estimates from limited calibration data.

Using this activation-dependent importance, the component-wise quantization cost of assigning bit-width b to $p_{e,k}^{(h)}$ combines the three factors in Eq. (8):

$$\mathcal{L}_{e,k}^{(h)}(b) \approx \underbrace{\left(\boldsymbol{\alpha}_{e,k}^{(h)}\right)^2}_{\substack{\text{Intrinsic spectral} \\ \text{importance}}} \cdot \underbrace{\boldsymbol{\beta}_{e,k}^{(h)}}_{\substack{\text{Activation-dependent} \\ \text{importance}}} \cdot \underbrace{\mathcal{E}_{e,k,b}^{(h)}}_{\substack{\text{Bit-width-dependent} \\ \text{distortion}}}. \tag{40}$$

The resulting sum of component-wise quantization costs is a normalized allocation surrogate; $\mathcal{L}_{e,P}^{(h)}(b)$ in the preceding derivation retains its original reconstruction-loss scale. Here $\mathcal{E}_{e,k,b}^{(h)} = \mathbb{E}_\epsilon\|\boldsymbol{\varepsilon}_{e,k,b}^{(h)}\|_2^2$ denotes the bit-width-dependent distortion. The following subsection specifies this factor for each candidate bit-width $b \in \mathcal{B}$.

C.3 Derivation of Bit-width-dependent Distortion

We derive bit-width-dependent distortion using a single-scale reference quantizer and use the resulting expressions as allocation surrogates for group-wise quantization.

The distortion derivation has the same form for every layer, expert, and projection type. Throughout App. C.3, we therefore omit the indices ℓ, e, and h, writing p_k and $\mathcal{E}_{k,b}$ for $p_{\ell,e,k}^{(h)}$ and $\mathcal{E}_{\ell,e,k,b}^{(h)}$, respectively, and applying the same convention to related quantities.

We now specify the bit-width-dependent distortion $\mathcal{E}_{k,b}$ incurred by quantizing the expert-specific spectral direction $p_k \in \mathbb{R}^m$, with $\|p_k\|_2 = 1$, at each candidate bit-width $b \in \mathcal{B} = \{16, 8, 6, 4, 3, 2, 1, 0\}$. The candidate $b = 16$ denotes an FP16 expert-specific spectral direction. Retaining the expert-specific spectral direction in FP16 incurs no additional quantization error, so $\mathcal{E}_{k,16} = 0$. As defined in Eq. (29), the bit-width-dependent distortion is

$$\mathcal{E}_{k,b} := \mathbb{E}_\epsilon\,\|\boldsymbol{\varepsilon}_{k,b}\|_2^2. \tag{41}$$

Bit-width-dependent distortion under the high-resolution approximation. For $b \in \{6, 8\}$, define

$$\rho_k := \|p_k\|_\infty, \qquad \eta_k := \frac{m\rho_k^2}{3}. \tag{42}$$

Symmetric uniform quantization is applied elementwise with a single quantization scale $s_{k,b}$ for all coordinates of p_k. For each coordinate $j \in \{1, \ldots, m\}$, define

$$Q_b(p_{k,j}; s_{k,b}) := s_{k,b} \cdot \mathrm{clamp}\left(\left\lfloor \frac{p_{k,j}}{s_{k,b}} \right\rceil, q_{\min}, q_{\max}\right), \qquad q_{\max} = 2^{b-1} - 1, \qquad q_{\min} = -2^{b-1}. \tag{43}$$

Here, $\lfloor z \rceil$ denotes rounding z to the nearest integer.

The coordinate-wise quantization error is

$$\varepsilon_{k,j,b} := p_{k,j} - Q_b(p_{k,j}; s_{k,b}). \tag{44}$$

Accordingly, the vector-level quantization error is

$$\boldsymbol{\varepsilon}_{k,b} := p_k - Q_b(p_k; s_{k,b}) = [\varepsilon_{k,1,b}, \ldots, \varepsilon_{k,m,b}]^\top. \tag{45}$$

In the high-resolution regime, clipping is negligible and each scalar rounding error is approximated as uniformly distributed on $[-s_{k,b}/2, s_{k,b}/2)$. Therefore, for each coordinate j,

$$\mathbb{E}_\epsilon[\varepsilon_{k,j,b}] = 0, \qquad \mathbb{E}_\epsilon\left[\varepsilon_{k,j,b}^2\right] = \frac{1}{s_{k,b}} \int_{-s_{k,b}/2}^{s_{k,b}/2} \varepsilon^2 d\varepsilon = \frac{s_{k,b}^2}{12}. \tag{46}$$

Since the vector-level squared error is the sum of coordinate-wise squared errors, we have

$$\begin{aligned} \mathbb{E}_\epsilon\left[\|\boldsymbol{\varepsilon}_{k,b}\|_2^2\right] &= \mathbb{E}_\epsilon\left[\sum_{j=1}^m \varepsilon_{k,j,b}^2\right] \\ &= \sum_{j=1}^m \mathbb{E}_\epsilon\left[\varepsilon_{k,j,b}^2\right] = \frac{m s_{k,b}^2}{12}. \end{aligned} \tag{47}$$

By the definition of ρ_k in Eq. (42), the quantization scale is

$$s_{k,b} = \frac{\rho_k}{q_{\max}}. \tag{48}$$

Substituting Eq. (48) into Eq. (47) gives

$$\mathbb{E}_\epsilon\left[\|\boldsymbol{\varepsilon}_{k,b}\|_2^2\right] = \frac{m\rho_k^2}{12 q_{\max}^2}. \tag{49}$$

For sufficiently large bit-widths, $q_{\max} = 2^{b-1} - 1 \approx 2^{b-1}$. With $\lambda := 2\ln 2$ and η_k defined in Eq. (42), we obtain

$$\begin{aligned} \mathcal{E}_{k,b} = \mathbb{E}_\epsilon\left[\|\boldsymbol{\varepsilon}_{k,b}\|_2^2\right] &\approx \frac{m\rho_k^2}{12 \cdot 2^{2(b-1)}} \\ &= \frac{m\rho_k^2}{3} \exp(-2b\ln 2) = \eta_k \exp(-\lambda b). \end{aligned} \tag{50}$$

Empirical bit-width-dependent distortion at low bit-widths. For $b \in \{2, 3, 4\}$, we use the symmetric uniform quantizer

$$Q_b(p_{k,j}; s_{k,b}) := s_{k,b} \cdot \mathrm{clamp}\left(\left\lfloor \frac{p_{k,j}}{s_{k,b}} \right\rceil, q_{\min}, q_{\max}\right), \tag{51}$$

where $q_{\max} = 2^{b-1} - 1$ and $q_{\min} = -2^{b-1}$. The vector quantizer $Q_b(p_k; s_{k,b})$ is applied element-wise. For a given scale $s_{k,b}$, the empirical bit-width-dependent distortion is

$$\mathcal{E}_{k,b}(s_{k,b}) := \|p_k - Q_b(p_k; s_{k,b})\|_2^2 = \sum_{j=1}^m \left(p_{k,j} - Q_b(p_{k,j}; s_{k,b})\right)^2. \tag{52}$$

For each pair of expert-specific spectral direction and bit-width, we choose the MSE-optimal quantization scale

$$s_{k,b}^* \in \arg\min_{s_{k,b}>0} \|p_k - Q_b(p_k; s_{k,b})\|_2^2. \tag{53}$$

At the selected scale $s_{k,b}^*$, define the coordinate-wise quantization error for $j \in \{1, \ldots, m\}$ as

$$\varepsilon_{k,j,b} := p_{k,j} - Q_b(p_{k,j}; s_{k,b}^*). \tag{54}$$

Evaluating the distortion at this scale gives

$$\mathcal{E}_{k,b} = \mathcal{E}_{k,b}(s^*_{k,b}) = \sum_{j=1}^{m} \varepsilon^2_{k,j,b} = m \left(\frac{1}{m} \sum_{j=1}^{m} \varepsilon^2_{k,j,b} \right). \tag{55}$$

We define the component-specific relative distortion ratio of p_k as

$$\kappa_{k,b} := \frac{\frac{1}{m}\sum_{j=1}^{m} \varepsilon^2_{k,j,b}}{\frac{1}{m}\sum_{j=1}^{m} p^2_{k,j}}. \tag{56}$$

Since p_k has unit norm, we have

$$\frac{1}{m} \sum_{j=1}^{m} p^2_{k,j} = \frac{1}{m}. \tag{57}$$

Combining Eq. (56), Eq. (55), and Eq. (57) gives

$$\kappa_{k,b} = \frac{\frac{1}{m}\sum_{j=1}^{m} \varepsilon^2_{k,j,b}}{\frac{1}{m}\sum_{j=1}^{m} p^2_{k,j}} = m \left(\frac{1}{m} \sum_{j=1}^{m} \varepsilon^2_{k,j,b} \right) = \mathcal{E}_{k,b}. \tag{58}$$

Since p_k has unit norm, its component-specific relative distortion ratio equals its total squared quantization error.

To calibrate a shared low-bit coefficient independently of model weights, let $z \sim \mathcal{N}(0, I_{d_{\text{ref}}})$ and $u = z/\|z\|_2$, where d_{ref} denotes the fixed dimension of the random vectors used to initialize κ_b. We define the shared low-bit coefficient as the expected MSE-optimal distortion of these random unit vectors:

$$\kappa_b := \mathbb{E}_z \left[\min_{s>0} \|u - Q_b(u; s)\|_2^2 \right]. \tag{59}$$

Repeated independent draws of u, each with its own MSE-optimal scale, give a Monte Carlo estimate of κ_b. Since u has unit norm, each squared error in Eq. (59) is also its relative distortion. Averaging these errors provides a model-independent starting point for the shared low-bit coefficient. Under the empirical coefficient-sharing approximation, we use the same κ_b as the low-bit surrogate for every spectral component across models, layers, experts, and projection types:

$$\mathcal{E}_{k,b} \approx \kappa_b, \qquad b \in \{2, 3, 4\}. \tag{60}$$

The random-vector calibration of κ_b is detailed in App. D.1; layer-wise and model-wide distortion diagnostics are reported in App. D.3.

Bit-width-dependent distortion for 1-bit sign quantization. For $b = 1$, we adopt the symmetric 1-bit sign quantizer

$$Q_1(p_k; s_{k,1}) := s_{k,1} \operatorname{sign}(p_k), \qquad s_{k,1} := \frac{1}{m} \sum_{j=1}^{m} |p_{k,j}|. \tag{61}$$

Define the normalized sign direction

$$\boldsymbol{r}_{k,1} := \frac{\operatorname{sign}(p_k)}{\sqrt{m}}, \qquad \cos\theta_k := p_k^\top \boldsymbol{r}_{k,1}, \tag{62}$$

where $\operatorname{sign}(\cdot)$ is applied elementwise with $\operatorname{sign}(0) = 1$ so that $\boldsymbol{r}_{k,1} \in \{\pm 1/\sqrt{m}\}^m$ and $\|\boldsymbol{r}_{k,1}\|_2 = 1$.

The quantized spectral direction at $b = 1$ in Eq. (61) can be rewritten using the normalized sign direction as

$$Q_1(p_k; s_{k,1}) = s_{k,1} \operatorname{sign}(p_k) = s_{k,1}\sqrt{m}\, \boldsymbol{r}_{k,1}. \tag{63}$$

By the definition of $\boldsymbol{r}_{k,1}$, its alignment with p_k is

$$\cos\theta_k = p_k^\top \boldsymbol{r}_{k,1} = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} p_{k,j} \operatorname{sign}(p_{k,j}) = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} |p_{k,j}| = s_{k,1}\sqrt{m}. \tag{64}$$

Therefore, the 1-bit reconstruction is equivalently

$$Q_1(p_k; s_{k,1}) = \cos\theta_k \boldsymbol{r}_{k,1}. \tag{65}$$

Thus the 1-bit quantization error vector is

$$\boldsymbol{\varepsilon}_{k,1} := p_k - Q_1(p_k; s_{k,1}) = p_k - \cos\theta_k \boldsymbol{r}_{k,1}. \tag{66}$$

Since $\|p_k\|_2 = \|\boldsymbol{r}_{k,1}\|_2 = 1$, the bit-width-dependent distortion at $b = 1$ is

$$\begin{aligned} \mathcal{E}_{k,1} &= \|\boldsymbol{\varepsilon}_{k,1}\|_2^2 = \|p_k - \cos\theta_k \boldsymbol{r}_{k,1}\|_2^2 \\ &= \|p_k\|_2^2 + \cos^2\theta_k \|\boldsymbol{r}_{k,1}\|_2^2 - 2\cos\theta_k p_k^\top \boldsymbol{r}_{k,1} \\ &= 1 + \cos^2\theta_k - 2\cos^2\theta_k \\ &= 1 - \cos^2\theta_k = \sin^2\theta_k. \end{aligned} \tag{67}$$

Bit-width-dependent distortion for zero-bit eviction. For $b = 0$, the expert-specific spectral direction is evicted:

$$Q_0(p_k) := \mathbf{0}. \tag{68}$$

The resulting zero-bit error vector is

$$\boldsymbol{\varepsilon}_{k,0} := p_k - Q_0(p_k) = p_k. \tag{69}$$

Taking the squared ℓ_2 norm gives the bit-width-dependent distortion at $b = 0$:

$$\mathcal{E}_{k,0} = \|\boldsymbol{\varepsilon}_{k,0}\|_2^2 = \|p_k\|_2^2 = 1, \tag{70}$$

where the last equality follows from the unit norm of the expert-specific spectral direction.

Piecewise bit-width-dependent distortion. Combining the high-bit approximation in Eq. (50), the low-bit surrogate in Eq. (60), the 1-bit and zero-bit distortions in Eq. (67) and Eq. (70), and the FP16 reference $\mathcal{E}_{k,16} = 0$ yields the bit-width-dependent distortion used by BITSMOE:

$$\mathcal{E}_{k,b} \approx \begin{cases} 0, & b = 16, \\ \eta_k \exp(-\lambda b), & b \in \{6, 8\}, \\ \kappa_b, & b \in \{2, 3, 4\}, \\ \sin^2\theta_k, & b = 1, \\ 1, & b = 0. \end{cases} \tag{71}$$

C.4 GLOBAL BIT BUDGET BY PROJECTION TYPE

Following the notation table (Tab. 5), $m = d_{\text{out}}$ denotes the dimension of each expert-specific spectral direction, and n denotes the number of spectral components per expert and projection type. In the full-component setting used here, $n = n^{(h)} = d_{\text{in}}$ for all $h \in \mathcal{H}$. We consider L MoE layers with the same expert count E and dimensions m, n. Each routed expert has three projection matrices with mn entries each. The target equivalent bit-width $\mathfrak{b}_{\text{eq}}$ therefore gives the total storage budget

$$M_{\text{budget}} = 3LEmn\mathfrak{b}_{\text{eq}} \quad \text{bits}. \tag{72}$$

Elsewhere in the paper, we report this target as the bit-width; for example, a 2-bit configuration uses $\mathfrak{b}_{\text{eq}} = 2$ under this storage accounting.

Each layer has one shared spectral basis per projection type, quantized to 8-bit precision using GPTQ. With all components retained, each basis has shape $n \times n$. The total shared-basis storage across all layers and projection types is therefore

$$M_{\text{share}} = 8 \cdot 3Ln^2 \quad \text{bits}. \tag{73}$$

Subtracting the shared-basis storage in Eq. (73) from the total storage budget in Eq. (72) gives the global bit budget for the expert-specific spectral directions:

$$B = M_{\text{budget}} - M_{\text{share}} = 3LEmn\mathfrak{b}_{\text{eq}} - 8 \cdot 3Ln^2 \quad \text{bits}. \tag{74}$$

We split this global budget equally across the three projection types, assigning one third to each:

$$B_{\text{bit}}^{(h)} = \frac{B}{3} = LEmn\mathfrak{b}_{\text{eq}} - 8Ln^2, \qquad h \in \mathcal{H}. \tag{75}$$

For each projection type, bits are allocated jointly across all layers under this global budget. For each layer ℓ and projection type h, the shared-basis decomposition produces n expert-specific spectral directions $\{p_{\ell,e,k}^{(h)}\}_{k=1}^{n}$ for each expert, with $p_{\ell,e,k}^{(h)} \in \mathbb{R}^m$. Assigning bit-width b to $p_{\ell,e,k}^{(h)}$ consumes mb bits. We therefore normalize the global bit budget for projection type h by m and obtain

$$B_{\text{alloc}}^{(h)}(\mathfrak{b}_{\text{eq}}) := \left\lfloor \frac{B_{\text{bit}}^{(h)}}{m} \right\rfloor = \left\lfloor LEn\mathfrak{b}_{\text{eq}} - \frac{8Ln^2}{m} \right\rfloor. \tag{76}$$

The resulting $B_{\text{alloc}}^{(h)}(\mathfrak{b}_{\text{eq}})$ is the global normalized bit budget for projection type h, corresponding to $B_{\text{alloc}}(\mathfrak{b}_{\text{eq}})$ in Eq. (10). For each type, we minimize the global FQCM loss in Eq. (9) through a single ILP with the budget constraint

$$\sum_{\ell=1}^{L}\sum_{e=1}^{E}\sum_{k=1}^{n}\sum_{b\in\mathcal{B}} b\, y_{\ell,e,k,b}^{(h)} \le B_{\text{alloc}}^{(h)}(\mathfrak{b}_{\text{eq}}), \tag{77}$$

where $y_{\ell,e,k,b}^{(h)} \in \{0,1\}$ indicates whether expert-specific spectral direction $p_{\ell,e,k}^{(h)}$ is assigned bit-width b, with

$$\sum_{b\in\mathcal{B}} y_{\ell,e,k,b}^{(h)} = 1, \qquad \forall \ell \in [L],\ e \in [E],\ h \in \mathcal{H},\ k \in [n]. \tag{78}$$

C.5 Component-wise Loss and ILP Formulation

The component-wise quantization cost in Eq. (40) combines three factors. Restoring the layer index ℓ, we define them as follows.

(1). Intrinsic spectral importance. The squared component magnitude $\left(\boldsymbol{\alpha}_{\ell,e,k}^{(h)}\right)^2 = \|\widetilde{p}_{\ell,e,k}^{(h)}\|_2^2$ is the intrinsic spectral importance, measuring the calibration-agnostic strength of the SSD spectral component.

(2). Activation-dependent importance. The activation-dependent importance $\boldsymbol{\beta}_{\ell,e,k}^{(h)}$ measures the component's sensitivity to calibration activations under the routing-weighted activation metric. For allocation, we use the normalized diagonal approximation described in App. C.2.

(3). Bit-width-dependent distortion. The bit-width-dependent distortion $\mathcal{E}_{\ell,e,k,b}^{(h)} := \mathbb{E}_\epsilon \|\boldsymbol{\varepsilon}_{\ell,e,k,b}^{(h)}\|_2^2$ measures the expected squared quantization error of the expert-specific spectral direction, where $\boldsymbol{\varepsilon}_{\ell,e,k,b}^{(h)} = p_{\ell,e,k}^{(h)} - Q_b(p_{\ell,e,k}^{(h)})$. Its values for the candidate bit-widths are modeled by Eq. (71).

Combining these factors, the component-wise quantization cost of assigning bit-width b to $p_{\ell,e,k}^{(h)}$ is approximated by

$$\mathcal{L}_{\ell,e,k}^{(h)}(b) \approx \underbrace{\left(\boldsymbol{\alpha}_{\ell,e,k}^{(h)}\right)^2}_{\substack{\text{Intrinsic spectral}\\ \text{importance}}} \cdot \underbrace{\boldsymbol{\beta}_{\ell,e,k}^{(h)}}_{\substack{\text{Activation-dependent}\\ \text{importance}}} \cdot \underbrace{\mathcal{E}_{\ell,e,k,b}^{(h)}}_{\substack{\text{Bit-width-dependent}\\ \text{distortion}}}. \tag{79}$$

For each projection type $h \in \mathcal{H}$, let $y_{\ell,e,k,b}^{(h)} \in \{0,1\}$ indicate whether spectral component (ℓ, e, h, k) is assigned bit-width $b \in \mathcal{B}$, and let y collect these assignment variables. As in Sec. 3.4, we define the global FQCM loss by summing the selected component-wise quantization costs over all L MoE layers:

$$\mathcal{L}_{\text{FQCM}}^{(h)}(y) := \sum_{\ell=1}^{L}\sum_{e=1}^{E}\sum_{k=1}^{n}\sum_{b\in\mathcal{B}} \mathcal{L}_{\ell,e,k}^{(h)}(b) y_{\ell,e,k,b}^{(h)}. \tag{80}$$

Given the global normalized bit budget $B^{(h)}_{\mathrm{alloc}}(\mathfrak{b}_{\mathrm{eq}})$ in Eq. (76), we minimize this global FQCM loss through a single ILP, following Eq. (10):

$$\begin{aligned} \text{MINIMIZE} \quad & \mathcal{L}^{(h)}_{\mathrm{FQCM}}(y) \\ \text{S.T.} \quad & \sum_{b\in\mathcal{B}} y^{(h)}_{\ell,e,k,b} = 1, \qquad \forall\, \ell \in [L],\ e \in [E],\ k \in [n], \\ & \sum_{\ell=1}^{L}\sum_{e=1}^{E}\sum_{k=1}^{n}\sum_{b\in\mathcal{B}} b\, y^{(h)}_{\ell,e,k,b} \le B^{(h)}_{\mathrm{alloc}}(\mathfrak{b}_{\mathrm{eq}}), \\ & y^{(h)}_{\ell,e,k,b} \in \{0,1\}, \qquad \forall\, \ell \in [L],\ e \in [E],\ k \in [n],\ b \in \mathcal{B}. \end{aligned} \tag{81}$$

We solve this global ILP independently for each projection type, allocating bits jointly across all MoE layers within its budget.

D Empirical Validation of Shared Distortion Coefficients

We examine the empirical basis for the shared low-bit distortion coefficients κ_b defined in Eq. (59). We describe their random-vector calibration and assess the dispersion of measured distortions in expert-specific spectral directions, both within layers and across all MoE layers for each projection type.

D.1 Calibration of Low-bit Distortion Coefficients

Following the empirical bit-width-dependent distortion formulation at low bit-widths in App. C.3, we estimate κ_b for $b \in \{2,3,4\}$ using independent Gaussian vectors $z_i \sim \mathcal{N}(0, I_{d_{\mathrm{ref}}})$ normalized as $u_i = z_i/\|z_i\|_2$. This normalization matches the unit norm of the expert-specific spectral directions.

For each vector and bit-width, we use the quantizer in Eq. (51), with integer range $[-2^{b-1}, 2^{b-1}-1]$ and one scale for the complete vector. The scale minimizes the empirical squared error:

$$s^*_{i,b} \in \arg\min_{s>0} \|u_i - Q_b(u_i; s)\|_2^2, \qquad D_{i,b} := \|u_i - Q_b(u_i; s^*_{i,b})\|_2^2. \tag{82}$$

We repeat this sampling-and-quantization procedure and average the resulting squared errors to estimate the reference coefficient in Eq. (59):

$$\widehat{\kappa}_b = \frac{1}{N}\sum_{i=1}^{N} D_{i,b}, \tag{83}$$

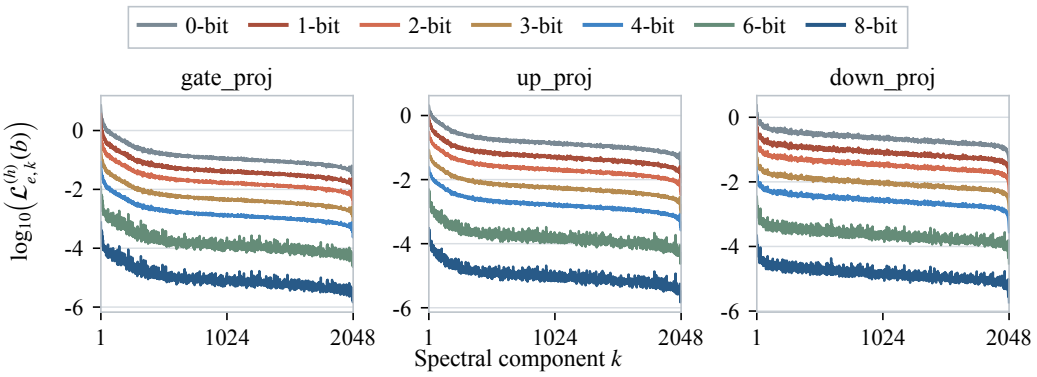
where N denotes the number of random draws. Since $\|u_i\|_2 = 1$, $D_{i,b}$ is also the relative distortion and is not divided by the vector dimension. This calibration provides a model-independent starting point for κ_b. Tab. 7 lists the shared low-bit distortion coefficients used in the model experiments; applying each coefficient across models, layers, experts, projection types, and spectral components follows the empirical coefficient-sharing approximation in Eq. (60).

Table 7: Shared low-bit distortion coefficients κ_b used in the model experiments.

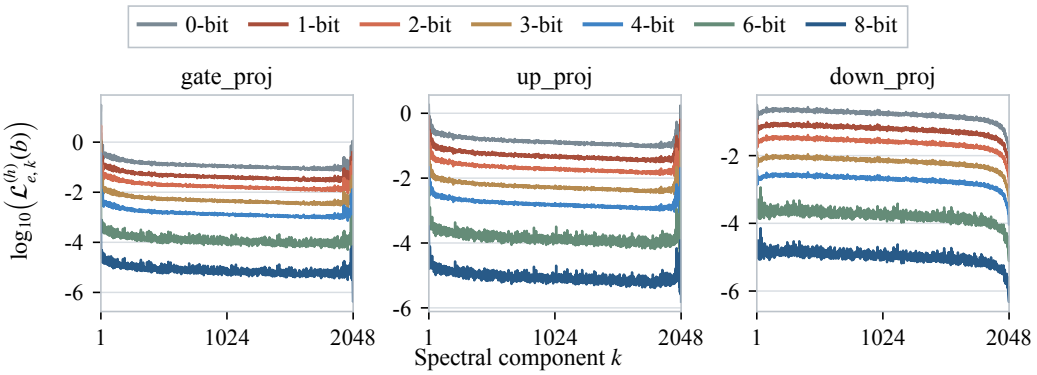
Bit-width b	4-bit	3-bit	2-bit
κ_b	0.01184786	0.04067890	0.14949200

D.2 Analysis of Component-wise Quantization Costs

Fig. 8 shows the component-wise quantization costs $\mathcal{L}^{(h)}_{e,k}(b)$ defined in Eq. (40) for selected layers and experts in DeepSeek-V2-Lite. These costs combine intrinsic spectral importance $\left(\boldsymbol{\alpha}^{(h)}_{e,k}\right)^2$, activation-dependent importance $\boldsymbol{\beta}^{(h)}_{e,k}$, and bit-width-dependent distortion $\mathcal{E}^{(h)}_{e,k,b}$, illustrating how the modeled quantization penalty varies with both the spectral component and the assigned precision.



(a) Layer 1, Expert 1.



(b) Layer 23, Expert 64.

Figure 8: Component-wise quantization costs $\mathcal{L}_{e,k}^{(h)}(b)$ across spectral components and bit-widths for selected layers and experts in DeepSeek-V2-Lite.

Fig. 8b also shows that the component-wise quantization costs for up_proj do not decrease monotonically with the spectral component index k: some components near the tail have substantially higher costs than those in the middle of the spectrum. The underlying statistics show increased activation-dependent importance $\beta_{e,k}^{(h)}$ at these tail components, offsetting their smaller intrinsic spectral importance $\left(\alpha_{e,k}^{(h)}\right)^2$. For a given spectral component, higher bit-widths generally yield lower component-wise quantization costs, consistent with the reduced bit-width-dependent distortion.

D.3 Dispersion of Bit-width-dependent Distortion

The following layer-wise statistics characterize measured bit-width-dependent distortions of model spectral directions separately from the random-vector calibration in App. D.1. For each layer ℓ and bit-width b, we average the bit-width-dependent distortions across all projection types, experts, and spectral components:

$$\overline{\mathcal{E}}_{\ell,b} = \frac{1}{E\sum_{h\in\mathcal{H}} n^{(h)}} \sum_{h\in\mathcal{H}} \sum_{e=1}^{E} \sum_{k=1}^{n^{(h)}} \left\| p_{\ell,e,k}^{(h)} - Q_b(p_{\ell,e,k}^{(h)}) \right\|_2^2, \tag{84}$$

where E is the number of routed experts, $\mathcal{H}$ is the set of projection types, and $n^{(h)}$ is the number of spectral components for projection type h. We use the coefficient of variation (CV) to quantify the relative dispersion of bit-width-dependent distortions within each layer:

$$\mathrm{CV}_{\ell,b} = \frac{s_{\ell,b}}{\left|\overline{\mathcal{E}}_{\ell,b}\right|} \times 100\%, \tag{85}$$

where $s_{\ell,b}$ denotes the sample standard deviation of the bit-width-dependent distortions over all projection types, experts, and spectral components in layer ℓ. Fig. 9 reports the layer-wise CV under 2-bit, 3-bit, and 4-bit quantization on DeepSeek-V2-Lite. The CV values are almost all below 15%, indicating low relative dispersion in the measured bit-width-dependent distortions.

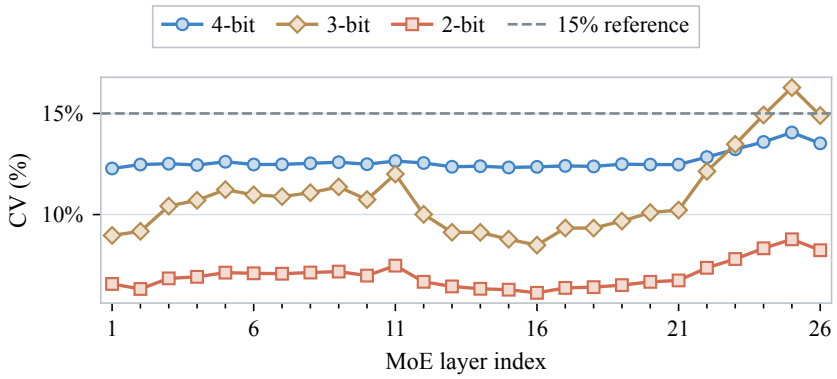


Figure 9: Layer-wise CV of measured bit-width-dependent distortions on DeepSeek-V2-Lite.

To match the scope of the global ILP, which allocates bits across all MoE layers independently for each projection type, we additionally pool the measured bit-width-dependent distortions across all MoE layers of DeepSeek-V2-Lite. For each projection type and bit-width, we compute the CV as the pooled sample standard deviation divided by the pooled mean, expressed as a percentage. This statistic includes both within-layer dispersion and differences between layer means.

Table 8: CV (%) of bit-width-dependent distortions pooled across all MoE layers, experts, and spectral components for each projection type in DeepSeek-V2-Lite.

Projection type	2-bit	3-bit	4-bit
gate_proj	8.3215	11.2069	12.5066
up_proj	5.9475	10.0072	12.5263
down_proj	6.6764	12.1075	12.9585

Tab. 8 shows that the pooled CV values remain below 13% for all three projection types and bit-widths. The values range from 5.95% to 8.32% at 2-bit, from 10.01% to 12.11% at 3-bit, and from 12.51% to 12.96% at 4-bit. These results indicate low relative dispersion in measured bit-width-dependent distortions across the full set of components considered by each global ILP.

E Additional Accuracy Results and Sensitivity Analysis

This section presents additional experimental results: task-level accuracy under ultra-low-bit quantization (App. E.1), task-level accuracy for the ablation studies (App. E.2), and sensitivity to calibration data composition (App. E.3).

E.1 Accuracy under Ultra-Low-Bit Quantization

We evaluate BitsMoE on Qwen3-30B-A3B and Qwen3-Next-80B-A3B at 1.4–2.0-bit precision. Tab. 9 reports task-level and average accuracy, with FP16 references and 2-bit HQQ/GPTQ baselines. For Qwen3-Next-80B-A3B, we additionally compare against 2-bit MiLo and MxMoE.

BitsMoE remains competitive with 2-bit baselines even at 1.4-bit. On Qwen3-30B-A3B, it achieves an average accuracy of 45.07%, exceeding 2-bit HQQ's 43.18% by 1.89 percentage points. On Qwen3-Next-80B-A3B, 1.4-bit BitsMoE achieves 64.14% average accuracy, surpassing 2-bit MxMoE's 51.14% by 13.00 percentage points. Although both methods use ILP-based mixed-precision bit allocation, BitsMoE achieves higher average accuracy in this comparison with a lower target bit-width.

The GSM8K results further demonstrate the preservation of mathematical reasoning performance below 2-bit precision. At 1.6-bit, BITSMOE retains 54.06% GSM8K accuracy on Qwen3-30B-A3B, exceeding 2-bit HQQ's 22.67% by 31.39 percentage points. On Qwen3-Next-80B-A3B, it achieves 66.79% at 1.6-bit, slightly above 2-bit HQQ's 66.49% by 0.30 percentage points. These results further support the effectiveness of BITSMOE for generation tasks requiring multi-step mathematical reasoning under constrained model-storage budgets.

Table 9: Task-level accuracy of BITSMOE on Qwen3-30B-A3B at 1.4–2.0-bit and Qwen3-Next-80B-A3B at 1.4–2.0-bit.

Model	Method	Bits	Accuracy (%) ↑							
			HS	AE	MQ	MM	OB	WG	GS	Avg.
QWEN3-30B-A3B	FP16	16	81.35	79.80	60.03	78.77	45.00	72.85	83.47	71.61
	HQQ	2	63.05	37.33	33.17	48.82	36.40	60.85	22.67	43.18
	GPTQ	2	70.16	59.97	24.89	39.17	39.40	60.62	4.32	42.65
	BITSMOE	2.0	**70.43**	**80.18**	**52.86**	**64.36**	**41.00**	**70.17**	**71.04**	**64.29**
		1.9	68.57	77.99	51.22	61.78	38.60	69.53	66.87	62.08
		1.8	66.21	76.14	48.41	60.40	37.00	67.40	64.75	60.04
		1.7	63.00	74.03	46.43	54.94	36.20	65.43	58.15	56.88
		1.6	60.57	73.70	43.45	48.18	35.00	65.04	54.06	54.29
		1.5	57.16	72.47	39.73	45.66	34.00	62.51	43.29	50.69
		1.4	51.08	68.18	36.68	39.63	31.80	57.54	30.55	45.07
QWEN3-NEXT-80B-A3B	FP16	16	82.72	84.64	63.85	84.53	44.20	76.80	77.18	73.42
	HQQ	2	**78.73**	83.50	50.95	79.68	43.80	70.40	66.49	67.65
	GPTQ	2	70.24	**84.30**	27.47	54.63	38.60	65.59	18.35	51.31
	MiLo	2	78.69	79.17	49.75	79.76	43.80	71.74	71.49	67.77
	MxMoE	2	80.68	67.26	24.96	65.20	44.40	74.51	0.99	51.14
	BITSMOE	2.0	78.05	83.92	**61.71**	**79.95**	**44.60**	**74.90**	**71.80**	**70.70**
		1.9	77.45	83.42	61.74	79.11	44.40	75.22	70.28	70.23
		1.8	76.37	83.04	59.80	77.93	43.80	74.59	69.07	69.23
		1.7	75.07	84.09	58.46	77.13	44.00	74.74	69.60	69.01
		1.6	73.39	83.63	56.52	75.97	42.60	74.19	66.79	67.58
		1.5	71.47	83.04	55.64	73.69	42.60	73.24	65.05	66.39
		1.4	68.66	82.24	53.50	71.39	39.40	71.35	62.47	64.14

E.2 SUPPLEMENTARY ABLATION RESULTS

Tab. 10 provides task-level results for the ablations on Qwen3-30B-A3B at 2-bit, complementing the average accuracies in Tab. 4. Uniform assigns equal budgets to all experts and quantizes spectral components to 2-bit in descending spectral energy order until each expert's budget is exhausted, assigning the remaining components to 0-bit. Consistent with Sec. 4.3, SSD with uniform allocation yields limited gains over independent SVD, whereas incorporating activation-dependent importance and routing weights substantially improves accuracy. Combining intrinsic spectral importance, activation-dependent importance, and routing weights achieves the highest average accuracy, supporting their complementary roles in component-wise mixed-precision bit allocation.

The task-level results also clarify the trade-off between shared-basis precision and the budget available for expert-specific spectral directions. On HellaSwag, a 4-bit shared basis achieves 79.71% accuracy, exceeding the 70.43% achieved with an 8-bit shared basis. However, the 8-bit shared basis performs better on the other six tasks and improves average accuracy from 62.78% to 64.29%. In particular, on GSM8K, which requires generating solutions to multi-step mathematical problems, it improves accuracy from 65.28% to 71.04%, a gain of 5.76 percentage points. At the same target equivalent bit-width, reducing shared-basis precision to 4-bit frees more of the storage budget for expert-specific spectral directions, but this additional budget does not translate into higher average accuracy. This pattern is consistent with the concern that quantization errors in the shared transformations can offset the benefits of allocating more bits to expert-specific spectral directions. These results support an 8-bit shared basis as a more favorable balance between shared-basis precision and expert-specific storage under the evaluated setting.

Table 10: Task-level accuracy for SSD, FQCM, and shared-basis precision ablations on Qwen3-30B-A3B at 2-bit. The Bits column specifies shared-basis storage precision.

Basis	Bits	Alloc.	α^2	β	G_e	Accuracy (%) ↑							
						HS	AE	MQ	MM	OB	WG	GS	Avg.
Non-Basis	–	Uniform	–	–	–	26.50	24.87	19.87	23.24	28.40	50.51	0.61	24.86
SSD	8-bit	Uniform	–	–	–	29.31	28.45	19.53	23.03	26.40	50.20	0.00	25.27
		FQCM	✗	✗	–	30.34	30.39	21.11	23.05	26.60	50.67	1.06	26.17
			✓	✗	–	32.13	32.24	19.77	23.56	26.80	50.28	0.61	26.48
			✗	✓	✗	54.29	71.38	33.70	55.10	33.20	62.35	20.39	47.20
			✗	✓	✓	65.32	77.10	44.12	60.55	37.40	66.61	54.36	57.92
			✓	✓	✗	67.23	76.85	46.37	61.92	38.20	67.48	65.13	60.45
	FP16	FQCM	✓	✓	✓	76.47	67.04	49.28	61.27	38.20	67.80	66.34	60.91
	4-bit		✓	✓	✓	**79.71**	69.81	52.66	63.78	39.40	68.82	65.28	62.78
	8-bit		✓	✓	✓	70.43	**80.18**	**52.86**	**64.36**	**41.00**	**70.17**	**71.04**	**64.29**

E.3 Sensitivity to Calibration Data Composition

Tab. 11 examines calibration-domain sensitivity on Qwen3-30B-A3B at 2-bit by varying the mixture of C4 (Raffel et al., 2020) and MATH (Hendrycks et al., 2021b) while keeping the calibration set fixed at 1,024 samples.

C4-only calibration achieves the highest average accuracy of 64.29%, while the C4–MATH mixtures remain close at 64.11–64.27%. Calibration on C4 helps identify spectral components important for preserving general language modeling ability, while adding MATH can improve mathematical reasoning performance. The 50% C4–50% MATH mixture raises MathQA accuracy from 52.86% to 53.90% and GSM8K accuracy from 71.04% to 72.78% relative to C4-only calibration. However, using MATH alone further improves MathQA accuracy to 55.48% but lowers GSM8K accuracy to 70.20% and average accuracy to 63.01%. This decline suggests that mathematical inputs alone may not adequately capture the importance of spectral components needed for general language modeling, which also supports solving mathematical word problems. These results support retaining general-domain data in calibration, with C4-only calibration providing the best average accuracy among the evaluated settings.

Table 11: Task-level results for different C4–MATH calibration mixtures on Qwen3-30B-A3B under 2-bit quantization.

C4	MATH	Accuracy (%) ↑							
		HS	AE	MQ	MM	OB	WG	GS	Avg.
100%	0%	70.43	80.18	52.86	64.36	41.00	70.17	71.04	64.29
80%	20%	70.83	79.55	54.00	64.89	39.00	71.03	70.58	64.27
50%	50%	70.39	79.29	53.90	63.71	39.80	69.85	72.78	64.25
20%	80%	70.77	78.75	54.97	63.62	39.20	69.77	71.72	64.11
0%	100%	68.02	78.11	55.48	62.94	37.20	69.14	70.20	63.01

F Inference Efficiency and Offline Quantization Costs

This section presents efficiency measurements for BitsMoE: time to first token, decode throughput, and peak GPU memory (App. F.1), and offline quantization time with a per-stage breakdown (App. F.2).

F.1 Inference Efficiency and GPU Memory Usage

We evaluate inference efficiency on two devices, a single RTX A6000 (48 GB) and a single RTX 3090 (24 GB), using the same benchmark parameters summarized in Tab. 12. We compare GPTQ

and BITSMOE at 2-bit using prompts from a pool of 512 samples drawn from the English C4 training split (Raffel et al., 2020). Measurements are averaged over three successful runs to reduce the impact of run-to-run variability. Decode throughput is measured in generated tokens per GPU decode second, and memory denotes peak PyTorch-allocated GPU memory. GPTQ uses the Triton backend described in Sec. 4.4.

Table 12: Settings for the inference efficiency and GPU memory benchmarks on a single RTX A6000 (48 GB) or RTX 3090 (24 GB).

Setting	Value
GPU	1 RTX A6000-48GB 1 RTX 3090-24GB
Prompt dataset	C4-en
`max_samples`	512
`max_length`	256
`truncation`	`True`
`padding`	`max_length`
`max_new_tokens`	128
`do_sample`	`False`
`batch_size`	1, 2, 4, 8, 16, 32, 64, 128, 256
`batch_size` (streaming)	1

Fig. 10 reports peak GPU memory across batch sizes on a single RTX A6000, complementing the decode-throughput results in Fig. 7. BITSMOE and GPTQ have similar memory footprints on DeepSeek-V2-Lite and Qwen3-Next-80B-A3B. On Qwen3-30B-A3B, BITSMOE uses modestly more memory, with the gap increasing from 0.29 GB at batch size 1 to 2.40 GB at batch size 256. The additional storage supports packing and quantization metadata and intermediate buffers used during inference. In conjunction with the throughput gains in Fig. 7, this modest overhead represents a practical tradeoff among ultra-low-bit precision, inference speed, and GPU memory usage. Both methods run out of memory at batch size 256 on DeepSeek-V2-Lite and Qwen3-Next-80B-A3B, as indicated by the shaded regions.

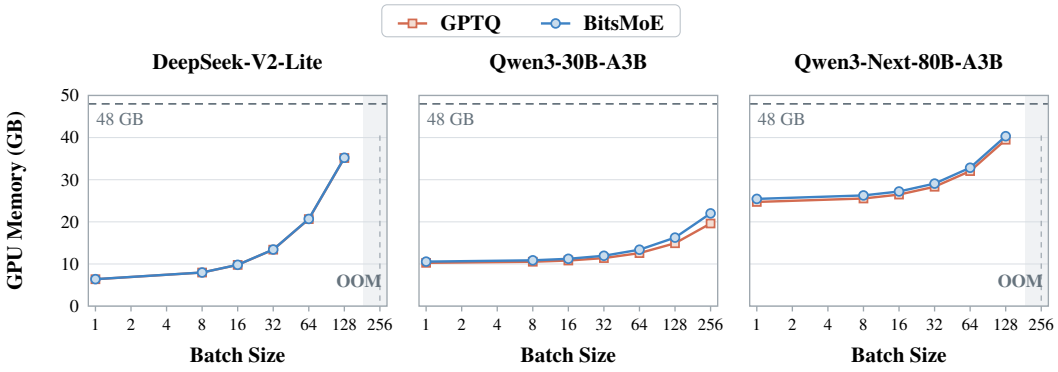


Figure 10: Peak GPU memory of GPTQ and BITSMOE across batch sizes.

We further evaluate streaming inference at batch size 1 on the RTX A6000, reporting time to first token (TTFT), decode throughput, and peak GPU memory in Tab. 13a. BITSMOE reduces TTFT and improves decode throughput across model scales, from DeepSeek-V2-Lite to Qwen3-Next-80B-A3B. For Qwen3-Next-80B-A3B, whose FP16 GPU memory footprint is 148.69 GB (Fig. 6b), the 2-bit model runs on a single 48 GB RTX A6000 with a peak allocation of 25.46 GB. It achieves 5.13 tok/s versus GPTQ's 2.44 tok/s, a 2.10× decode speedup, while reducing TTFT from 7.13 s to 0.99 s. These results demonstrate the value of ultra-low-bit deployment for serving large MoE models under single-GPU memory constraints.

On a single RTX 3090 with 24 GB of memory, BITSMOE also achieves lower TTFT and higher decode throughput than GPTQ on Qwen3-30B-A3B. Tab. 14 reports inference performance across

Table 13: Time to first token, decode throughput, and peak GPU memory at 2-bit for streaming inference at batch size 1 on a single RTX A6000 (48 GB) or RTX 3090 (24 GB).

(a) On RTX A6000 (48 GB).

Model	TTFT (s) ↓			Decode throughput (tok/s) ↑			Memory (GB) ↓	
	GPTQ	BITSMOE	Speedup	GPTQ	BITSMOE	Speedup	GPTQ	BITSMOE
DEEPSEEK-V2-LITE	1.57	0.61	**2.57**×	6.99	9.58	**1.37**×	6.39	6.40
QWEN3-30B-A3B	3.00	0.85	**3.53**×	3.08	7.19	**2.33**×	10.28	10.56
QWEN3-NEXT-80B-A3B	7.13	0.99	**7.20**×	2.44	5.13	**2.10**×	24.73	25.46

(b) On RTX 3090 (24 GB).

Model	TTFT (s) ↓			Decode throughput (tok/s) ↑			Memory (GB) ↓	
	GPTQ	BITSMOE	Speedup	GPTQ	BITSMOE	Speedup	GPTQ	BITSMOE
QWEN3-30B-A3B	2.32	0.84	**2.76**×	3.97	7.52	**1.89**×	10.28	10.56

batch sizes 1–256, while Tab. 13b reports streaming inference at batch size 1. Decode speedups range from $1.96\times$ to $4.95\times$, computed from the reported means before rounding. At batch size 256, BITSMOE reaches 242.61 tok/s versus GPTQ's 81.02 tok/s, a $2.99\times$ speedup, with peak allocated memory of 22.01 GB. For streaming inference, it reduces TTFT from 2.32 s to 0.84 s and increases decode throughput from 3.97 tok/s to 7.52 tok/s. Thus, a 30B-parameter MoE model can support both streaming generation and batched decoding on a single 24 GB GPU, extending the practical benefits of BITSMOE to resource-constrained deployment.

Table 14: Decode throughput and peak GPU memory of Qwen3-30B-A3B at 2-bit on a single RTX 3090 (24 GB).

Batch size	Decode throughput (tok/s) ↑			Memory (GB) ↓	
	GPTQ	BITSMOE	Speedup	GPTQ	BITSMOE
1	3.83	7.51	**1.96**×	10.27	10.56
2	4.35	12.62	**2.90**×	10.31	10.60
4	4.83	23.52	**4.87**×	10.39	10.69
8	6.54	32.37	**4.95**×	10.53	10.87
16	9.52	43.48	**4.57**×	10.82	11.23
32	14.66	70.12	**4.78**×	11.40	11.95
64	24.60	115.48	**4.69**×	12.58	13.39
128	43.70	183.21	**4.19**×	14.92	16.26
256	81.02	242.61	**2.99**×	19.61	22.01

F.2 OFFLINE QUANTIZATION TIME BREAKDOWN

We compare offline quantization time at 2-bit on a single NVIDIA A100 PCIe (80 GB). For all three models, GPTQ and BITSMOE use the same calibration set of 1,024 samples. Tab. 15 reports total time for both methods and breaks down the BITSMOE pipeline into calibration-statistics collection (Calib.), shared-basis construction (SVD), bit allocation (ILP), quantization (Quantize), bit-plane packing (Packing), and auxiliary processing (Other). All times are in minutes, with speedups over GPTQ shown in parentheses.

BITSMOE requires less offline quantization time than GPTQ on all three models. It quantizes DeepSeek-V2-Lite, Qwen3-30B-A3B, and Qwen3-Next-80B-A3B in 21.32, 63.41, and 267.04 minutes, respectively, compared with GPTQ's 59, 298, and 1,279 minutes. The corresponding speedups are $2.77\times$, $4.70\times$, and $4.79\times$, demonstrating lower preprocessing cost across the evaluated model scales.

Quantization remains the most time-consuming stage of BITSMOE. FQCM precomputes component-wise quantization costs, enabling global bit allocation through an ILP with a linear objective and linear constraints, without repeated empirical evaluation of candidate allocations. With

the same hardware and 1,024 calibration samples, BITSMOE quantizes Qwen3-Next-80B-A3B in 267.04 minutes, comparable to GPTQ's 298 minutes for Qwen3-30B-A3B. This comparison supports the practical scalability of our offline quantization pipeline to larger MoE models.

Table 15: Offline quantization time (min) at 2-bit on an NVIDIA A100 PCIe (80 GB) using 1,024 calibration samples, with per-stage costs for BITSMOE.

Model	GPTQ	BITSMOE						
		Calib.	SVD	ILP	Quantize	Packing	Other	Total
DEEPSEEK-V2-LITE	59.00	6.13	0.99	0.66	8.30	0.85	4.39	**21.32** (2.77×)
QWEN3-30B-A3B	298.00	11.68	1.93	4.50	29.90	3.19	12.21	**63.41** (4.70×)
QWEN3-NEXT-80B-A3B	1279.00	18.47	4.17	57.67	129.27	12.66	44.80	**267.04** (4.79×)

Table 16: Offline quantization time breakdown (min) of ILP-based methods on Qwen3-30B-A3B at 2-bit.

Method	Calibration	ILP solving	Quantization	Other	Total↓
MxMoE	2120.70	0.29	977.87	7.98	3106.83
GEMQ	69.10	0.29	326.73	648.28	1044.40
BITSMOE	11.68	4.50	29.90	17.33	**63.41**

Tab. 16 compares offline quantization time on Qwen3-30B-A3B at 2-bit. All methods are evaluated on NVIDIA A100-PCIe-80 GB GPUs using the same 1,024 C4 calibration samples, each with 2,048 tokens. MxMoE's Calibration stage jointly performs calibration, candidate quantization, and sensitivity-cost construction. GEMQ's Other stage includes the computation of gradient-weighted reconstruction errors for bit allocation.

As shown in Tab. 16, BITSMOE achieves $49.00\times$ and $16.47\times$ offline quantization speedups over the ILP-based methods MxMoE (Duanmu et al., 2025) and GEMQ (Deng et al., 2026), respectively. Although our ILP takes 4.50 minutes compared with 0.29 minutes for both baselines, the additional solver time is more than offset by savings in cost construction and quantization. MxMoE measures output perturbations for candidate quantization schemes, while GEMQ estimates gradient-weighted reconstruction costs using calibration data. FQCM computes component-wise quantization costs directly from intrinsic spectral importance, activation-dependent importance, and bit-width-dependent distortion, avoiding repeated evaluation of candidate output errors. In addition, BITSMOE quantizes expert-specific spectral directions directly without GPTQ's iterative Hessian-based error compensation. GPTQ is applied only to the shared bases at 8-bit precision. These choices help explain the lower cost-construction overhead and the shorter quantization time of 29.90 minutes, compared with 977.87 minutes for MxMoE and 326.73 minutes for GEMQ.