QuantMoE-Bench: Examining Post-Training Quantization for Mixture-of-Experts

Anonymous ACL submission

Abstract

Mixture-of-Experts (MoE) scales large language models by increasing parameters with nearly constant inference FLOPs via sparse activation, but incurs significant memory overhead. This necessitates model compression techniques. Post-training quantization offers a powerful approach for model compression. Existing methods adopt a fixed quantization precision for the entire MoE model. This rigid setup can lead to suboptimal performance, without considering the inherent sparse structure. For instance, consistently activated shared experts may require higher precision than selectively used token-conditioned experts. This paper investigates fine-grained, MoE structure-aware quantization, exploring heuristics from coarse (MoE layers) to fine (linear layers) granularity. Our extensive benchmarking on two MoE models across six tasks reveals a critical principle: different MoE structures require varying bit precisions for effective quantization. Our fine-grained mixed-precision approach achieves state-of-the-art average performance (65.35%) compared to baselines like GPTQ (64.30%). Based on these findings, we introduce novel data-driven bit allocation techniques, including an outlier-aware linear layer scorer and a block importance predictor.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success in various natural language processing tasks (OpenAI et al., 2024; Touvron et al., 2023). However, the rapid growth in model size, with state-of-the-art LLMs containing billions of parameters, poses significant challenges to computational resources and memory consumption (Aminabadi et al., 2022; Lin et al., 2024; Shoeybi et al., 2020). Mixture-of-Experts (MoE) (Shazeer et al., 2017) has emerged as a promising solution to address the computation overhead. MoE allows for the scaling up of LLMs while maintaining roughly constant FLOPs. By incorporating multiple experts and employing a sparse gating mechanism, MoE achieves efficient computation, enabling the development of larger models within the constraints of limited computation (Dai et al., 2024).

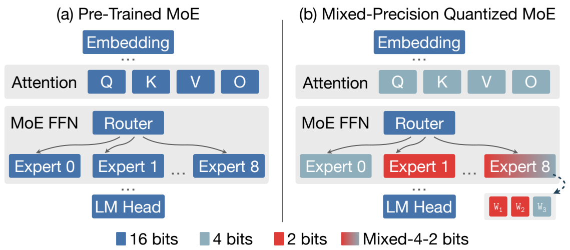


Figure 1: After our post-training quantization, the MoE (a) is quantized into (b) with mixed precisions.

Despite its advantages, MoE still suffers from extensive memory costs due to its vast parameter size, which hinders its practical deployment. For example, pretrained in 8-bit precision, the DeepSeek-V3 (Liu et al., 2024) MoE model takes around 1.3 TB memory while only 74 GB parameters are activated for each input token. Model compression techniques tailored to MoE architectures are essential to address this issue. Existing MoE compression methods can be categorized into two main approaches: merging and pruning. Expert merging, such as MC-MoE(Li et al., 2024), aims to reduce the memory footprint by combining similar experts based on routing policy and compressing the resulting model using low-rank decomposition. Expert pruning, such as task-specific pruning (Chen et al., 2022), focuses on identifying and removing the least important experts or connections based on their contribution to a specific task. However, these approaches not only necessitate prohibitively expensive model retraining but also operate under task-specific settings, limiting their practicality.

Post-training quantization is a family of model compression techniques that converts pre-trained model weights from high-precision formats (*e.g.*, FP32) to lower-precision representations without model retraining or task-specific tuning. Recent works, such as GPTQ (Frantar et al., 2023a), which adapts quantization intervals based on the Hessian information, and SmoothQuant (Lin et al., 2024), which jointly quantizes the model weight and activation by offline migrating the activation outliers, have demonstrated the effectiveness of post-training quantization for dense LLMs toward 4 bits compression. This advantage is particularly desired for MoE-based LLMs given their vast parameter sizes and prevailing deployment for various tasks (Jiang et al., 2024; Liu et al., 2024).

However, our experiments show that directly deploy those techniques for MoE models leads to subpar performance. Existing methods employ a uniform bit-width across all components of the MoE model. This one-size-fits-all approach fails to account for the inherent sparse structure of the MoE architecture. For example, the sparse expert activations in MoE models exhibit distinct statistical properties from dense activations, suggesting adaptive bit allocation among experts' quantization. This yields the primary **research question**:

(RQ) Do different MoE structures require varying numbers of bits for effective quantization?

Our investigations reveal that different components in MoE require varying bit allocations, as shown in Figure 1. For example, shared experts and the first few MoE layers demand higher precision for effective quantization. Moreover, these findings naturally motivate two key questions: (1) How to identify the layers that are more sensitive to quantization; (2) How to systematically determine the importance of each MoE layer for bit allocation. To address these questions, we introduce novel data-driven techniques for optimizing bit allocation in MoE quantization, including the outlier-aware linear layer scorer that captures weight magnitude variations, and the MoE block importance predictor that leverages block-level activations patterns. Our key contributions are listed:

1. We establish the first benchmark for Mixture-of-Experts post-training quantization, *i.e.*, `QuantMoE-Bench`. This benchmark encompasses investigations into four critical MoE-related heuristics by evaluating different quantization methods including GPTQ and SmoothQuant, and analyzes multiple bit allocation strategies on attention layers, FFNN layers, experts, and MoE blocks. Our evaluation covers two representative MoE LLMs and six benchmark tasks.
2. Our benchmark study uncovers critical MoE quantization principles: attention layers require higher precision than FFNNs, shared experts need more bits than token-conditioned experts (4-bit *v.s.* 2-bit), and earlier MoE layers demand higher precision compared to later ones. These insights enable optimal bit allocation under constrained memory budgets while maintaining model performance.
3. Through extensive experiments, we demonstrate that our fine-grained mixed precision quantization approach achieves state-of-the-art performance, improving average task performance by 1.05% compared to existing methods (GPTQ).
4. Leveraging the insights from our benchmark study, we introduce novel data-driven techniques to optimize bit allocation in MoE quantization. These include the development of outlier-aware linear layer scorer and MoE block importance predictor, which significantly improve the effectiveness of mixed-precision quantization by 0.97%.

2 Related Works

Mixture-of-Experts. Mixture-of-Experts (MoE) approach (Shazeer et al., 2017) enhances neural network scalability by using router networks to activate model segments according to input tokens selectively. Numerous efforts have adapted feed-forward neural networks (FFNNs) within Transformers to incorporate MoE layers, constructing MoE language models (Dai et al., 2024; Fedus et al., 2022; Jiang et al., 2024). Several variants, for example, DeepSeek-MoE (Dai et al., 2024) employs finely segmented experts and designates a select few as shared experts to capture common knowledge. MoE is widely acknowledged for its superior generative abilities and remarkable efficiency (Artetxe et al., 2022; Dai et al., 2024; Fedus et al., 2022; Jiang et al., 2024; Krajewski et al., 2024; Rajbhandari et al., 2022). The recent work Mixtral (Jiang et al., 2024) illustrates that MoE can match the performance of equivalent full-parameter LLMs while utilizing far fewer active parameters. However, MoE suffers from significant memory overhead, posing challenges to its efficient deployment (Li et al., 2024; Liu et al., 2024; Xue et al., 2024; Krajewski et al., 2024; Luo et al., 2024).

MoE Compression. Current works to reduce the memory overhead of MoE models mainly focus on reducing the number of experts. An earlier approach (Chen et al., 2022) involves pruning non-

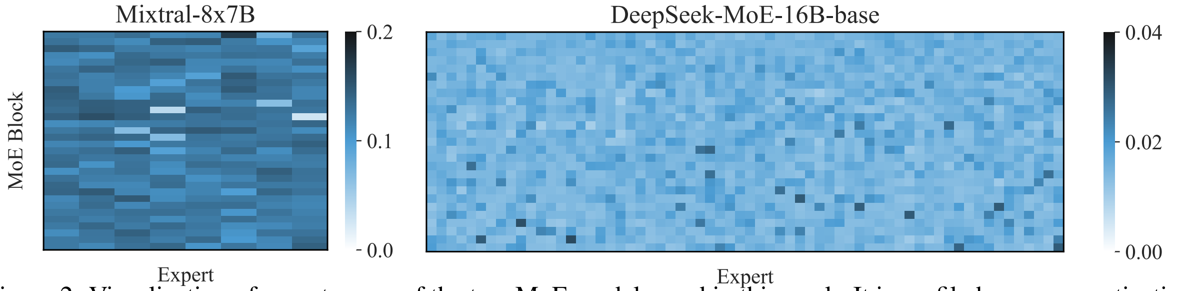


Figure 2: Visualization of expert usage of the two MoE models used in this work. It is profiled on our quantization calibration data, *i.e.*, 512 random 4096 token sequences from the WikiText dataset (Merity et al., 2016).

essential experts for a specific downstream task during fine-tuning, utilizing statistics based on cumulative usage frequency. MC-SMoE (Li et al., 2024) identifies and groups similar experts, subsequently merging them and further decomposing the merged expert into low-rank components within each group. However, these approaches are developed under task-specific fine-tuning settings and do not explore the development of the MoE compression towards general LLMs.

Post-Training Quantization. Post-training quantization reduces computational and storage demands by converting pre-trained models from high-precision to lower-precision formats without extensive retraining (Frantar et al., 2023b,a). It has been widely applied to LLMs, optimizing them for deployment on resource-constrained devices. Techniques like layer-wise quantization and mixed-precision schemes are designed for minimal performance degradation while reducing model size and computational requirements (Liu et al., 2023; Pan et al., 2023; Sharify et al., 2024). Recent methods such as SmoothQuant (Xiao et al., 2024), GPTQ (Frantar et al., 2023a), AWQ (Lin et al., 2024), and address specific challenges for LLMs. GPTQ (Frantar et al., 2023a) employs layer-wise and mixed-precision quantization to balance efficiency and accuracy. AWQ (Lin et al., 2024) adapts to weight sensitivity, preserving critical weights' precision while aggressively quantizing less sensitive ones. These advancements in PTQ enable significant reductions in LLM computing and storage while preserving performance. However, their effectiveness on MoE models is underexplored.

3 Preliminary

3.1 Quantization Method

The primary objective of this work is to benchmark several MoE-related heuristics combined with established LLM quantization techniques. Given that the substantial memory overhead of MoE models predominantly originates from their weights, we adopt GPTQ (Frantar et al., 2023a), a popular weight quantization method. GPTQ executes layer-by-layer weight quantization by addressing a specific reconstruction problem for each layer. Specifically, let $\mathbf{W}$ be the weights of a linear layer and $\mathbf{X}$ be the input to that layer derived from a small subset of calibration data, the reconstruction problem is defined as:

$$\texttt{argmin}_{\widehat{\mathbf{W}}}, ||\mathbf{W}\mathbf{X} - \widehat{\mathbf{W}}\mathbf{X}||_2^2$$

This objective, being the sum of squared errors, forms a quadratic equation, allowing the greedy-optimal update of weights to be calculated element-by-element using the Hessian information, $\mathbf{H} = 2\mathbf{X}\mathbf{X}^\top$. GPTQ further enhances this process by incorporating a lazy-batch update and a Cholesky reformulation, to improve scalability and numerical stability for LLM quantization.

3.2 Mixture-of-Experts

There are several variants of MoE in the context of LLMs, such as attention MoE and FFNN MoE. In this work, we explore the quantization of MoE models that utilize router networks to selectively activate FFNNs for different input tokens. Specifically, for the i-th expert's feed-forward function at the l-th transformer layer, denoted as $\texttt{FFNN}i^l(\cdot)$, the output of the MoE layer for the input hidden states $\mathbf{X}$ is given by:

$$\texttt{FFNN}^l_{\text{MoE}}(\mathbf{X}) = \sum_{i=1}^{l} \mathcal{G}(\mathbf{W}_l\mathbf{X}) \cdot \texttt{FFNN}^l_i(\mathbf{X}),$$

where $\mathbf{W}_l$ represents a linear routing matrix and $\mathcal{G}(\cdot)$ is a routing function that typically employs a top-k selection mechanism, resulting in a sparse output. Due to the duplication of FFNN layers, the principal memory overhead in the MoE model is attributed to the FFNN component.

3.3 Expert Usage as A Heuristic

As the routing of experts in MoE models is not ideally balanced, expert usage frequency and its variants have emerged as prevalent heuristics for measuring the importance of different experts within an MoE block (Chen et al., 2022; Li et al., 2024). For instance, task-specific expert pruning proposed by (Chen et al., 2022) uses a criterion based on cumulatively calculated expert routing probabilities for pruning during fine-tuning on a specific task. In this paper, focusing on post-training quantization, we utilize the routing distribution from the calibration data as the heuristic for expert usage. Specifically, for the l-th MoE block, equipped with a routing matrix $\mathbf{W}_l \in \mathbb{R}^{e \times d}$ and input hidden states $\mathbf{X} \in \mathbb{R}^{b \times d}$ from the calibration data, the expert usage heuristic is:

$$\texttt{usage} = \texttt{normalize}\left(\sum_i \mathcal{G}(\mathbf{W}_l \mathbf{X}_i)\right),$$

where $\mathcal{G}(\cdot)$ is the routing function employing a top-k selection mechanism that yields a sparse binary output. We visualize the calculated expert usage of Mixtral-8x7B and DeepSeek-MoE-16B-base MoE models on the quantization calibration data, as shown in Figure 2. Note that Mixtral-8x7B demonstrates a more balanced routing distribution than DeepSeek-MoE-16B-base.

4 Benchmark Post-Quantization for MoE

In this section, we present several heuristics for MoE quantization and the empirical performance of them. Our benchmarking covers two MoE models and six popular tasks.

4.1 Benchmark Setups

MoE Models. We select two representative MoE models for our benchmark evaluation, *i.e.*, Mixtral-8x7B (Jiang et al., 2024) and DeepSeek-MoE-16B-base (Dai et al., 2024). Mixtral-8x7B substitutes every FFNN with a MoE block and has 8 experts per MoE block with top-2 routing, while DeepSeek-MoE-16B-base uses a fine-grained MoE architecture by including 64 experts with top-6 routing and 2 shared experts per MoE block.

Quantization. We mainly focus on *weight-only mixed-precision* quantization, we further extend our experiments and conclusions to its combination with activation quantization in Section 5. We use GPTQ (Frantar et al., 2023a) for quantization, without loss of generality. Throughout this work, we use a group size of 128. Our experiments emphasize an extreme quantization scenario, where most weights are quantized to either 2 or 4 bits.

Calibration and Evaluation Details. We use the calibration data consisting of 512 random 4096 token sequences from the WikiText dataset (Merity et al., 2016), following GPTQ (Frantar et al., 2023a). Unlike previous literature that focuses on language modeling benchmarks (Xiao et al., 2024; Lin et al., 2024; Frantar et al., 2023a), we evaluate all the methods on six popular LLM tasks for a practical benchmarking: WinoGrande (ai2, 2019), COPA (Gordon et al., 2012), OpenBookQA (OBQA) (Mihaylov et al., 2018), HellaSwag (Zellers et al., 2019), and MMLU (Hendrycks et al., 2021). We report the performance on MMLU with 5-shot and all others with zero-shot. All experiments are conducted with PyTorch on 8 NVIDIA H100s, and we utilize *lm-evaluation-harness* [1] for the evaluation.

4.2 Problem Formulation

In this section, we formalize the intrinsic tradeoff between model performance and bit allocation in MoE quantization as Pareto optimization.

Given an MoE model with n weight components (attention layers, MoE FFNN experts, *etc.*), we aim to find the optimal bit allocation strategy that balances model performance and memory efficiency. Let $\mathbf{b} = (b_1, b_2, \ldots, b_n)$ represent a bit allocation vector, where $b_i \in \{2, 4\}$ is the quantization precision for the i-th component. We formulate this as a bi-objective optimization problem:

$$\begin{aligned} \max_{\mathbf{b}} \quad & \mathcal{P}(\mathbf{b}) \\ \min_{\mathbf{b}} \quad & \mathcal{B}(\mathbf{b}) = \frac{1}{n}\sum_{i=1}^{n} b_i \\ \text{subject to} \quad & b_i \in \{2, 4\}, \quad \forall i \in \{1, 2, \ldots, n\}, \end{aligned}$$

where $\mathcal{P}(\mathbf{b})$ is the model performance and $\mathcal{B}(\mathbf{b})$ is the average bit-width across all components.

Given the combinatorial nature of this problem and the difficulty in obtaining a closed-form expression for $\mathcal{P}(\mathbf{b})$, we approach it using the structure-aware heuristics described in the following sections, leveraging our observations about the varying importance of MoE components. As shown in Figure 3, our mixed-precision approach effectively navigates the Pareto frontier, offering superior performance compared to uniform bit allocation.

4.3 Benchmark Results

We first evaluate the four bit-allocation heuristic MoE quantization approaches using GPTQ on Mixtral-8x7B and DeepSeek-MoE-16B.

Table 1 presents the overall comparison performance of our mixed-precision MoE quantization strategies *v.s.* baseline (*i.e.* GPTQ) across

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

Methodology	Bits	WinoGrande (%)	COPA (%)	OBQA (%)	HellaSwag (%)	PIQA (%)	MMLU (%)	Average (%)
				Mixtral-8x7B				
GPTQ	3	73.56	93.00	44.80	75.31	79.11	58.37	70.69
GPTQ	2	49.33	63.00	25.40	28.18	52.99	24.29	40.53
+Attn	2.06	54.14	65.00	26.80	44.87	57.78	27.38	46.00 ↑5.47
+Attn+Freq	3.27	73.09	92.00	44.80	77.03	78.29	63.05	71.38 ↑30.85
+Attn+Freq+FirstL	3.51	73.32	95.00	44.00	80.06	79.71	64.89	72.83 ↑32.30
+Attn+LinearOSP	3.12	72.85	92.00	43.40	78.14	78.40	62.92	71.29 ↑30.76
+Attn+LayerISP	3.35	71.19	93.00	44.00	77.03	78.29	63.15	71.11 ↑30.58
				DeepSeek-MoE-16B-base				
GPTQ	3	67.87	87.00	40.40	72.92	78.50	39.12	64.30
GPTQ	2	53.27	76.00	30.20	45.32	66.53	25.28	49.43
+Attn	2.06	61.72	83.00	34.00	61.93	73.45	26.53	56.77 ↑7.34
+Attn+Shared	2.12	65.67	85.00	39.20	67.84	76.28	35.28	61.55 ↑12.12
+Attn+Shared+Freq	3.00	68.82	88.00	42.00	73.32	77.31	41.51	65.16 ↑15.90
+Attn+Shared+Freq+FirstL	3.06	68.35	88.00	42.60	73.85	77.69	41.58	65.35 ↑15.92
+Attn+Shared+LinearOSP	3.06	67.32	88.00	41.60	72.45	77.48	41.64	64.84 ↑15.41
+Attn+Shared+LayerISP	3.06	69.30	87.00	42.00	73.56	77.86	41.86	65.27 ↑15.67

Table 1: Comparison of quantization bit allocation strategies including attention (Attn) and shared expert (Shared) prioritization, frequency-based expert selection (Freq), first-layer prioritization (FirstL), linear outlier score predictor (LinearOSP) and layer importance score predictor (LayerISP). We evaluate performance on Mixtral-8x7B and DeepSeek-MoE-16B-base across multiple benchmarks. Unlike uniform-precision quantization (*i.e.* GPTQ), our approach shows superiority by tailoring bit allocation to the MoE structure.

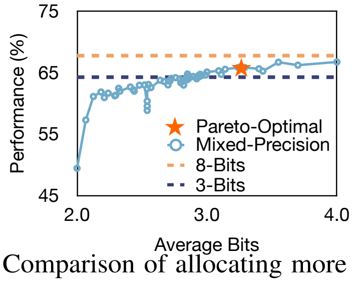


Figure 3: Comparison of allocating more bits (*i.e.* 4 bits) for attention and frequent experts with uniform-bits quantization. The Pareto-optimal bit is 3.29.

multiple benchmarks Mixtral-8x7B and DeepSeek-MoE-16B-base. The results highlight that uniform quantization leads to substantial performance degradation, reinforcing the need for structure-aware, fine-grained quantization strategies. By incorporating methods such as attention-aware adjustments (+Attn), expert frequency-based bit allocation (+Freq), and prioritization of early MoE layers (+FirstL), importance-based quantization (+LinearOSP and +LayerISP), we observe improvements in model performance, particularly in lower-bit settings. A mixed-precision approach tailored to the MoE structure outperforms fixed-bit quantization. In particular, expert usage frequency-based quantization (+Freq) improves performance by dynamically allocating more bits to frequently used experts, though its impact varies per the routing balance of models. Prioritizing the first few MoE blocks (+FirstL) yields better results. Lastly, importance-based strategies(+LinearOSP, +LayerISP) further refine bit allocation.

To gain deeper insights, we further systematically analyze four key research questions: (1) the effectiveness of expert usage frequency as a quantization heuristic, (2) whether attention or FFNN layers deserve more bit precision, (3) the importance of the first versus last MoE blocks in quantization, and (4) the necessity of allocating more bits to shared experts. The following sections provide a detailed discussion of each aspect.

Q1: Is expert usage frequency a good quantization heuristic? A: Fairly good. Expert usage frequency is a popular heuristic in the compression of MoE models, predicated on the insight that less frequently used experts are likely less crucial. Our experiments in Table 2 show its effectiveness as a quantization heuristic for MoE. Specifically, for the DeepSeek model, this heuristic outperforms randomly allocating more bits to experts, likely due to the model's unbalanced routing distribution. However, for the Mixtral model, where routing distribution is more balanced, the advantage of using expert usage frequency is less significant.

Q2: Attention *vs.* FFNN: Which Deserves More Bits in MoE? A: Attention layers are more bit-efficient. Because of the unique characteristics of the FFNN within the MoE framework, we compare the attention layer and the FFNN layer to determine which deserves more bits. We compare the performances by quantizing the attention layers with more bits *v.s.* randomly selecting experts in the FFNN layers with more bits, with the same average bits of the entire MoE model for a fair comparison. Specifically, we quantize the attention or randomly selected FFNN weight to {2, 4,

Methodology	Bits	WinoGrande (%)	COPA (%)	OBQA (%)	HellaSwag (%)	PIQA (%)	MMLU (%)	Average (%)
				`Mixtral-8x7B`				
Random 2	2.54	**58.59 ± 2.57**	68.00 ± 11.27	**33.00 ± 1.78**	46.60 ± 18.21	60.14 ± 9.32	28.26 ± 4.64	49.10 ± 7.73
Frequent 2	2.54	58.33	**76.00**	32.00	**56.62**	**66.21**	**36.01**	**54.20**
Random 4	3.03	67.77 ± 0.36	**86.33 ± 3.51**	38.47 ± 0.31	67.48 ± 0.52	**73.99 ± 0.52**	48.13 ± 2.57	63.70 ± 0.49
Frequent 4	3.03	**68.82**	86.00	**38.80**	**67.68**	72.20	**49.42**	**63.82**
				`DeepSeek-MoE-16B-base`				
Random 10	2.53	**67.28 ± 0.04**	**88.50 ± 1.50**	38.40 ± 0.80	**70.99 ± 0.50**	**76.74 ± 0.84**	35.23 ± 0.09	62.86 ± 0.60
Frequent 10	2.53	66.46	87.00	**39.60**	70.31	76.71	**37.84**	**62.99**
Random 15	2.68	**67.25 ± 0.47**	84.50 ± 2.50	**40.00 ± 0.60**	**71.79 ± 0.43**	76.85 ± 0.08	35.71 ± 0.82	62.68 ± 0.71
Frequent 15	2.68	67.17	**88.00**	39.00	71.09	**76.93**	**40.59**	**63.80**
Random 20	2.83	67.25 ± 0.47	84.50 ± 2.50	40.00 ± 0.60	71.79 ± 0.43	76.85 ± 0.08	35.71 ± 0.82	62.68 ± 0.71
Frequent 20	2.83	**67.25**	**86.00**	**40.40**	**72.06**	**77.58**	**40.78**	**64.01**
Random 25	2.97	67.72 ± 0.24	89.00 ± 1.00	**40.70 ± 0.10**	71.98 ± 0.19	77.04 ± 0.05	36.54 ± 1.55	63.83 ± 0.04
Frequent 25	2.97	**67.72**	**90.00**	39.20	**72.83**	**77.15**	**41.06**	**64.66**

Table 2: Comparison of the expert usage frequency heuristic *v.s.* random allocation. For the `Mixtral-8x7B` model, we compare the allocation of 4 bits to the top-$\{2, 4\}$ most frequently used experts per MoE block against randomly selecting $\{2, 4\}$ experts for the same bit allocation. For the `DeepSeek-MoE-16B-base` model, we keep shared expert $\{8\}$ bits and compare between top-$\{10, 15, 20, 25\}$ most frequently used experts against randomly selecting $\{10, 15, 20, 25\}$ experts per MoE block. Other experts are quantized to 2 bits, while all attention layers are uniformly quantized to 4 bits. We provide the mean value a and standard deviation b over 3 independent trials as $a \pm b$..

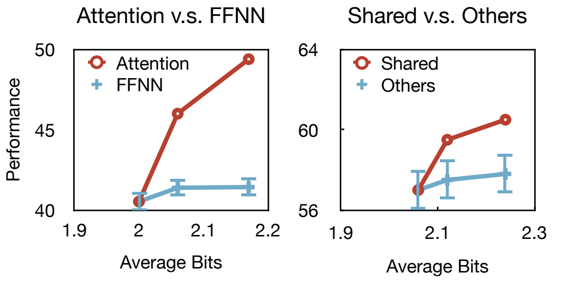


Figure 4: Comparison of quantizing more bits for attention *v.s.* FFNN and shared experts *v.s.* others, evaluated on Mixtral model. FFNN and others' results show the mean and standard deviation from 3 independent trials.

8} bits, while all other weights are quantized to 2 bits. As illustrated in Figure 4 (left), quantizing attention weights to more bits (*i.e.*, 4 or 8 bits) consistently results in significant performance gains (over 5%) under each average bit allocation for the MoE model. This greater efficiency likely stems from the fact that attention weights are activated for every token, while FFNN weights only engage with a subset of input tokens. Consequently, increasing the quantization bits for FFNN weights does not benefit all inputs. Based on these findings, attention weights are quantized to 4 bits by default in all following experiments.

Q3: Do the model's first or last MoE blocks deserve more bits in quantization? A: The first MoE blocks. We investigate which layer of the MoE block is more critical and thus deserves more bits during for quantization. As shown in Table 3, we evaluate the performance of allocating more bits to the first k blocks *v.s.* the last k blocks in quantization. The results consistently indicate that higher bit quantization of the first few blocks yields better performance, suggesting that we can allocate more bits to the quantization of the first blocks. This aligns with prior studies that empirically show the greater importance of first few Transformer blocks (Dai et al., 2024; Ma et al., 2023).

Q4: Does the shared expert always deserve more bits? A: Yes. The DeepSeek MoE model includes two shared experts in each MoE block to capture common knowledge across domains. To evaluate their role in quantization, we compare allocating more bits to the two shared experts *v.s.* randomly selecting two non-shared experts, with the same average bits for a fair comparison. The shared or random non-shared experts are quantized to 2, 4, 8 bits, while attention weights are set to 4 bits and all other weights to 2 bits. As shown in Figure 4 (right), allocating more bits (*i.e.*, 4 or 8 bits) to shared experts consistently yields superior performance. This is attributed to the shared experts being activated for every input token, unlike non-shared experts that only engage with subsets.

5 Extended Study

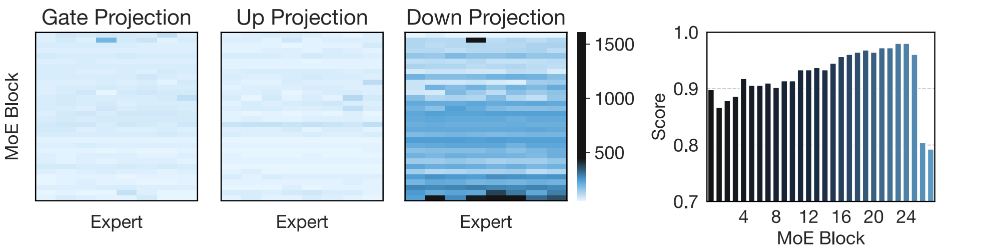
In this section, we introduce two novel data-driven techniques aimed at identifying crucial components in MoE to improve quantization performance.

5.1 Outlier-Aware Linear Layer Scorer

Insight. From the quantization perspective, the larger the range of a weight magnitude group, the more difficult it will be for quantization. We found

Methodology	Bits	WinoGrande (%)	COPA (%)	OBQA (%)	HellaSwag (%)	PIQA (%)	MMLU (%)	Average (%)
				`Mixtral-8x7B`				
First 4	2.30	**57.85**	**72.00**	**32.80**	**52.80**	**61.59**	**29.65**	**51.12**
Last 4	2.30	53.75	60.00	27.80	46.25	58.87	26.56	45.54
First 8	2.54	**62.11**	**85.00**	**35.80**	**62.72**	**67.74**	**35.61**	**58.16**
Last 8	2.54	52.09	69.00	29.60	47.87	59.58	26.03	47.36
				`DeepSeek-MoE-16B-base`				
First 4	2.29	**65.27**	**85.00**	**38.40**	**64.42**	72.74	**28.88**	**59.12**
Last 4	2.29	62.90	83.00	36.00	64.41	**74.65**	27.38	58.06
First 8	2.63	**64.09**	**86.00**	**38.75**	**67.84**	75.35	30.12	**60.36**
Last 8	2.63	62.83	83.00	37.80	65.94	**75.73**	**31.00**	59.38

Table 3: Comparison between quantizing first k *v.s.* last k MoE blocks with higher (*i.e.* 4) bits. All weights in attention layers are quantized to 4 bits, and the other weights are quantized to 2 bits. In `DeepSeek-MoE-16B-base` model, we keep the first block that is dense block as 4 bits by default. We evaluate k of 4 and 8. The higher performance of each comparison pair is marked as **bold**.



(a) Outlier-Aware Linear Layer Scorer

(b) MoE Block Importance Score Predictor

Figure 5: (a) Visualization of the `outlier-score` metric applied to each FFNN linear weight matrix in the Mixtral model. We present the *gate projection* (left), *up projection* (middle), and *down projection* (right) in FFNN experts separately. (b) Visualization of the `MoE block importance score` metric applied on the DeepSeek MoE model.

that, in MoE, each FFNN linear weight matrix consists predominantly of values within a narrow range, interspersed with a few significant outliers. Consequently, we propose a weight-magnitude-based metric to identify those linear layers that are challenging to quantize effectively, thereby necessitating a higher allocation of quantization bits.

Methodology. We define the metrics to estimate the outliers of weights by the maximum ratio of the largest to the average absolute magnitude within each column. Specifically, for a weight matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$, we compute this metric as follows:

$$\texttt{outlier-score}(\mathbf{W}) = \texttt{max}_j \left(\frac{\texttt{max}(|\mathbf{W}:, j|)}{\texttt{mean}(|\mathbf{W}:, j|)} \right),$$

With this metric, we can identify those linear layers that require more quantization bits and allocate more to them, providing an effective trade-off between performance and efficiency. The overall procedure is detailed in Algorithm 1.

Experiments. We evaluate this metric by comparing its application for the top-$p\%$ of linear layers against randomly selecting linear layers, using percentages of 25% and 50%. As shown in Table 4, our proposed scorer consistently outperforms the random baseline on both models and almost all tasks (except HellaSwag and MMLU). This is particularly evident in the DeepSeek model, where it achieves an average performance improvement of about 3%, aligning with our expectations.

Visualization. As shown in Figure 5 (a), we visualize the proposed `outlier-score` for each FFNN linear weight of the Mixtral model. Given that each FFNN expert includes three linear layers, *i.e. gate projection*, *up projection*, and *down projection*, we visualize these components separately. Notably, many of the *down projection* linear layers, particularly later layers in the model, exhibit significantly higher `outlier-scores` compared to others.

5.2 MoE Block Importance Score Predictor

Inspired by Q3 in Section 4.3, which demonstrates that allocating more bits to different MoE blocks yields variable performance improvements, we propose a novel method to identify and quantize those critical blocks with additional bits.

Insight. We find an increasing cosine similarity between the tensors generated before and after FFN blocks for some of the MoE blocks, indicating less important computing output produced by these blocks. This observation aligns with observations on dense models in previous literature (Jaiswal et al., 2024). Therefore, the basic idea is that less accurate output of these blocks producing tokens with high cosine similarity will not affect the over-

Methodology	Bits	WinoGrande (%)	COPA (%)	OBQA (%)	HellaSwag (%)	PIQA (%)	MMLU (%)	Average (%)
				`Mixtral-8x7B`				
Random 25%	2.54	60.74 ± 0.63	78.67 ± 4.62	34.07 ± 1.63	$\mathbf{57.36} \pm 0.53$	68.19 ± 0.74	$\mathbf{32.49 \pm 1.60}$	55.25 ± 0.95
Ours top-25%	2.54	**62.19**	**83.00**	**35.80**	57.04	**68.23**	30.95	**56.20**
				`DeepSeek-MoE-16B-base`				
Random 25%	2.54	64.04 ± 0.78	84.67 ± 4.73	37.53 ± 0.46	67.39 ± 0.71	74.61 ± 0.60	29.43 ± 1.31	59.61 ± 0.76
Ours top-25%	2.54	**66.14**	**85.00**	**38.80**	**71.65**	**76.82**	**36.19**	**62.43**

Table 4: Comparison between using our linear weight scorer *vs.* random selection of linear layers for bit allocation in quantization. We quantize 25% of linear layers across all MoE blocks to 4 bits. All attention weights are quantized to 4 bits, all other weights are quantized to 2 bits. In each comparison pair, the higher performance is highlighted in **bold**. We provide the mean value a and standard deviation b over 3 independent trials as $a \pm b$.

Methodology	Bits	WinoGrande (%)	COPA (%)	OBQA (%)	HellaSwag (%)	PIQA (%)	MMLU (%)	Average (%)
				`DeepSeek-MoE-16B-base`				
Random 4	2.29	61.09 ± 0.78	83.00 ± 0.00	37.20 ± 0.85	64.88 ± 0.30	74.21 ± 0.08	27.82 ± 0.46	58.03 ± 0.13
First 4	2.29	65.27	**85.00**	**38.40**	64.42	72.74	28.88	59.12
Predicted 4	2.29	**65.27**	83.00	36.60	**64.88**	**74.54**	**37.75**	**60.34**
Random 8	2.63	64.48 ± 0.83	85.33 ± 3.21	38.73 ± 0.95	67.57 ± 0.40	$\mathbf{75.43 \pm 0.14}$	$\mathbf{31.41 \pm 2.17}$	60.49 ± 0.56
First 8	2.63	64.09	**86.00**	**38.75**	67.84	75.35	30.12	60.36
Predicted 8	2.63	**65.35**	**86.00**	38.00	**68.77**	75.35	30.01	**60.58**
Random 12	2.92	64.64 ± 0.89	83.50 ± 0.71	$\mathbf{39.60} \pm 2.83$	69.51 ± 0.56	75.98 ± 0.42	32.57 ± 0.30	60.97 ± 0.62
First 12	2.92	67.48	**88.00**	38.60	70.59	75.95	**39.25**	63.31
Predicted 12	2.92	**68.11**	**88.00**	39.20	**71.82**	**76.66**	38.45	**63.71**

Table 5: Comparison between using our MoE block importance predictor *v.s.* two baselines: ①random selecting and ②first k MoE blocks. The predicted or selected MoE blocks are quantized to 4 bits, all attention weights are quantized to 4 bits, and all other weights are quantized to 2 bits. In each comparison, the highest performance is highlighted in **bold**. We provide the mean value a and standard deviation b over 3 independent trials as $a \pm b$.

all model performance much, thus lower weight bits might not hurt performance much.

Methodology. To capture the generalized hidden states' dynamic information of each MoE block, we train a two-layer FFNN with a tangent activation function. This network predicts the cosine similarity between the input and output hidden states. We train it on 400 sequences, each of 1024 tokens from WikiText (Merity et al., 2016). The training procedure is in Algorithm 2. During quantization, we apply this predictor and compute an average predicted score for each MoE layer across all tokens. **A higher score indicates less importance and fewer bits for quantization.**

Experiments. In Table 5, we compare the performance of using our block importance predictor to select k MoE blocks for 4 bits and others for 2 bits quantization with two other baselines: ① random selecting k MoE blocks, and ② first k MoE blocks (as it is the best in Q3 in Section 4.3). Evaluation results on the DeepSeek-MoE-16B-base model are presented in Table 5, showing the superiority of our method against the other two baselines.

Visualization. We visualize the predicted scores of MoE blocks using our trained predictors in the DeepSeek model, as shown in Figure 5 (b). Notably, MoE blocks in the middle of the model, which exhibit higher scores, are regarded as less critical. Consequently, these blocks are quantized to fewer bits (*i.e.*, 2 bits), reflecting their lower importance. Besides, Figure 5 (b) demonstrates that the first few MoE blocks are more important aligned with Q3. Interestingly, the last two blocks of the DeepSeek model are also crucial, thereby allocating more bits and yielding better performance.

6 Conclusion

This work introduces mixed-precision MoE post-training quantization (PTQ) through a systematic investigation of various heuristic-based approaches. While conventional quantization techniques (*e.g.*, GPTQ) show limited effectiveness when directly applied to MoE models, the question of optimal bit allocation across different MoE model components requires deeper exploration. We present `QuantMoE-Bench`, the first comprehensive benchmark to study PTQ of MoE models that reveals critical insights, including significant importance variations in the MoE model. Drawing on these insights, we further develop a block importance predictor and a linear layer outlier range scorer to more precisely identify critical components in quantization. Our methods significantly improve the effectiveness of MoE model quantization.

Limitation

We primarily focus on weight-only quantization GPTQ, without extensively exploring alternative quantization techniques (*e.g.* AWQ) or activation quantization. Our proposed approaches, while effective, lack theoretical guarantees. Our evaluation metrics prioritize task performance without real-world, efficient implementation.

References

2019. Winogrande: An adversarial winograd schema challenge at scale.

Reza Yazdani Aminabadi, Samyam Rajbhandari, Minjia Zhang, Ammar Ahmad Awan, Cheng Li, Du Li, Elton Zheng, Jeff Rasley, Shaden Smith, Olatunji Ruwase, and Yuxiong He. 2022. Deepspeed inference: Enabling efficient inference of transformer models at unprecedented scale. *Preprint*, arXiv:2207.00032.

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giri Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O'Horo, and 5 others. 2022. Efficient large scale language modeling with mixtures of experts. *Preprint*, arXiv:2112.10684.

Tianyu Chen, Shaohan Huang, Yuan Xie, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. 2022. Task-specific expert pruning for sparse mixture-of-experts. *Preprint*, arXiv:2206.00277.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *Preprint*, arXiv:2401.06066.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Preprint*, arXiv:2101.03961.

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

Elias Frantar, Sidak Pal Singh, and Dan Alistarh. 2023b. Optimal brain compression: A framework for accurate post-training quantization and pruning. *Preprint*, arXiv:2208.11580.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398, Montréal, Canada. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Ajay Jaiswal, Bodun Hu, Lu Yin, Yeonju Ro, Shiwei Liu, Tianlong Chen, and Aditya Akella. 2024. Ffn-skipllm: A hidden gem for autoregressive decoding with adaptive feed forward skipping. *Preprint*, arXiv:2404.03865.

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

Jakub Krajewski, Jan Ludziejewski, Kamil Adamczewski, Maciej Pióro, Michał Krutul, Szymon Antoniak, Kamil Ciebiera, Krystian Król, Tomasz Odrzygóźdź, Piotr Sankowski, Marek Cygan, and Sebastian Jaszczur. 2024. Scaling laws for fine-grained mixture of experts. *Preprint*, arXiv:2402.07871.

Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. 2024. Merge, then compress: Demystify efficient smoe with hints from its routing policy. *Preprint*, arXiv:2310.01334.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for llm compression and acceleration. *Preprint*, arXiv:2306.00978.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Shih-yang Liu, Zechun Liu, Xijie Huang, Pingcheng Dong, and Kwang-Ting Cheng. 2023. Llm-fp4: 4-bit floating-point quantized transformers. *arXiv preprint arXiv:2310.16836*.

Shuqing Luo, Jie Peng, Pingzhi Li, and Tianlong Chen. 2024. Hexa-moe: Efficient and heterogeneous-aware moe acceleration with zero computation redundancy. *Preprint*, arXiv:2411.01288.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Preprint*, arXiv:2305.11627.

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

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

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Jiayi Pan, Chengcan Wang, Kaifu Zheng, Yangguang Li, Zhenyu Wang, and Bin Feng. 2023. Smoothquant+: Accurate and efficient 4-bit post-training weightquantization for llm. *arXiv preprint arXiv:2312.03788*.

Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. 2022. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale. *Preprint*, arXiv:2201.05596.

Sayeh Sharify, Zifei Xu, Xin Wang, and 1 others. 2024. Combining multiple post-training techniques to achieve most efficient quantized llms. *arXiv preprint arXiv:2405.07135*.

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

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2020. Megatron-lm: Training multi-billion parameter language models using model parallelism. *Preprint*, arXiv:1909.08053.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2024. Smoothquant: Accurate and efficient post-training quantization for large language models. *Preprint*, arXiv:2211.10438.

Leyang Xue, Yao Fu, Zhan Lu, Luo Mai, and Mahesh Marina. 2024. Moe-infinity: Offloading-efficient moe model serving. *Preprint*, arXiv:2401.14361.

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

A Appendix

A.1 Evaluation Datasets

In this section, we introduce details of the datasets in our evaluation. For a more comprehensive study, we have selected six popular benchmark tasks: WinoGrande, COPA, OpenBookQA (OBQA), HellaSwag, and MMLU.

WinoGrande (ai2, 2019) is a large-scale dataset designed for commonsense reasoning, consisting of pronoun resolution problems. Each instance in the dataset presents a sentence with an ambiguous pronoun that needs to be resolved based on context. This task tests the model's ability to understand and reason about everyday situations.

The Choice of Plausible Alternatives (COPA) dataset (Gordon et al., 2012) focuses on causal reasoning. Each question in COPA consists of a premise and two choices, where the model must select the more plausible alternative. This task evaluates the model's understanding of cause-and-effect relationships in natural language.

OpenBookQA (Mihaylov et al., 2018) is a multiple-choice question-answering dataset that requires the model to use both scientific facts and commonsense knowledge. The dataset challenges the model's ability to combine factual knowledge with reasoning to answer questions correctly.

HellaSwag (Zellers et al., 2019) is a benchmark for commonsense NLI (Natural Language Inference) that tests the model's ability to predict the most plausible continuation of a given sentence. The dataset contains scenarios from various domains, such as cooking and sports, requiring the model to understand context and plausibility.

The Massive Multitask Language Understanding (MMLU) benchmark (Hendrycks et al., 2021) evaluates models across a wide range of subjects, from elementary mathematics to law. For this study, we report performance on MMLU with a 5-shot setting, where the model is given five examples per task before evaluation, allowing us to gauge the model's few-shot learning capabilities.

We perform a zero-shot evaluation on WinoGrande, COPA, OpenBookQA, and HellaSwag, where the model is not provided with any task-specific training examples. For MMLU, a 5-shot evaluation protocol is adopted, providing five examples per task. This setup helps us assess the generalization ability of the models across different types of reasoning and knowledge-based tasks.

A.2 Random Seed

For all the random selection experiments, we use random seeds $\{42, 43, 44\}$ to conduct three independent trials and then report the standard deviation and mean.

A.3 Further Discussion

In this section, we present further discussion of the DeepSeek-MoE-16B-base performance across different bits.

Expert usage frequency.As shown by Q1 in Section 4.3, expert usage frequency is a critical metric in the compression of MoE models, predicated on the insight that less frequently used experts are likely less crucial. We present further discussion of ablation on the bits allocation in the expert-frequency-based methods.

In Table 8, we compare the allocation of $\{4, 8\}$ bits of the selected top-k experts, while all other experts are quantized to 2 bits. We quantize the shared experts and attention weights to 8 bits. Table 8 indicates that increasing the bit width of frequently activated experts improves performance. However, the gain from increasing the top-k expert bits from 4 to 8 is minimal.

Combination of the weight outlier and expert usage frequency. We conducted additional experiments on the DeepSeek-MoE-16B-base model by integrating bit-width allocation based on layers with significant weight outliers with allocation based on expert usage frequency to explore the trade-off between them. Specifically, we aimed for a total average bit budget of 2.97. We select portions of the model to be quantized to 4 bits using a combination of the two heuristics, while quantizing all attention weights to 4 bits and all other weights to 2 bits. For selecting the 4-bit weights, we introduce a hyper-parameter, α (0 ¡ α ¡ 1), representing the proportion of weights chosen based on expert usage frequency, with the remainder selected based on weight outliers. We varies α to illustrate the trade-off between these methods, as detailed above. As shown in Table 7, the optimal combination of these two methods occurs when alpha is set to 0.1. This means that 20% of the 4-bit MoE weights are selected based on expert usage frequency, while the remaining 80% are chosen according to weight outliers.

Baseline results of low-precision quantization. We provide the 16-bit (FP16), 4-bit, and 2-bit baselines of both Mixtral-8x7B and DeepSeek-MoE-16B-base models in Table 9.

Algorithm 1 The Procedure of MoE Mixed-Precision Quantization with `outlier-score`.

```
Initialize: A MoE model with $l$ linear layers across all the FFNN experts, the number of linear layers for $4$ bit quantization $k$.
Let $\mathcal{M}$ and $\mathcal{S}$ represent the set of each linear layer matrix in FFNN and its score, respectively.
for linear layer $i = 1, \dots, l$ do
    $\mathbf{W} \leftarrow \mathcal{M}[i]$
    $\mathcal{S}[i] \leftarrow \mathtt{max}_j \left( \frac{\mathtt{max}(|\mathbf{W}_{:,j}|)}{\mathtt{mean}(|\mathbf{W}_{:,j}|)} \right)$
end for
$\alpha \leftarrow \mathtt{sorted}(\mathcal{S})[k]$
4bits-quantize $(\{\mathcal{M}[i] \mid \mathcal{S}[i] >= \alpha\})$
2bits-quantize $(\{\mathcal{M}[i] \mid \mathcal{S}[i] < \alpha\})$
Return: A quantized mixed-precision MoE model.
```

Algorithm 2 The Training Procedure of Block Score Predictor.

```
Initialize: A MoE block $M$, token input and output embedding set at block $M$ $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i \in [N]}$.
Let $\mathcal{BSP}$ denotes the block score predictor.
$\mathcal{X} \leftarrow \{\mathbf{x}_i \mid i \in [N]\}$
$\mathcal{S} \leftarrow \{\mathtt{cosine}(\mathbf{x}_i, \mathbf{y}_i) \mid i \in [N]\}$
$\mathcal{BSP} \leftarrow \mathtt{train}(\mathcal{X}, \mathcal{S})$
Return: The importance scorer $\mathcal{BSP}$ for MoE Block $M$.
```

Table 6: Baseline results of the 16-bit (FP16), 4-bit, and 2-bit quantization.

Bits	WinoGrande (%)	COPA (%)	OBQA (%)	HellaSwag (%)	PIQA (%)	MMLU (%)	Average (%)
Mixtral-8x7B							
16	76.48	93.00	47.00	83.98	82.37	70.35	75.33
4	74.98	92.00	46.20	81.65	80.85	67.65	73.89
2	49.33	63.00	25.40	28.18	52.99	24.29	40.53
DeepSeek-MoE-16B-base							
16	70.40	91.00	44.20	77.35	78.72	44.77	67.74
4	71.35	87.00	43.20	76.39	78.51	44.22	66.78
2	53.28	76.00	30.20	45.33	66.54	25.28	49.44

Table 7: The combination of weight outlier and expert usage frequency, evaluated on the DeepSeek-MoE-16B-base model.

Bits	α	WinoGrande (%)	COPA (%)	OBQA (%)	HellaSwag (%)	PIQA (%)	MMLU (%)	Average (%)
2.97	0.0	67.72	90.00	39.20	72.83	77.15	41.06	64.66
	0.1	68.11	89.00	41.60	72.88	77.80	41.84	65.21
	0.2	69.21	89.00	41.20	72.60	76.93	41.60	65.09
	0.3	68.92	88.00	42.00	72.06	76.65	41.21	64.81
	0.4	67.48	89.00	41.40	71.88	76.71	40.96	64.57
	0.5	67.32	90.00	40.80	71.89	76.93	40.21	64.52
	0.6	65.90	87.00	39.40	71.86	76.76	38.67	63.27
	0.7	66.21	87.00	41.40	71.45	76.87	36.98	63.32
	0.8	66.45	89.00	41.00	70.89	76.60	37.67	63.60
	0.9	66.37	84.00	40.20	70.83	76.87	39.84	63.02
	1.0	68.19	87.00	41.60	71.01	76.11	40.81	64.12

Table 8: Ablation on the allocated bits for the selected top-k experts based on frequency. We compare the allocation of $\{4, 8\}$ bits of the top-k experts based on frequency, and all other experts are quantized to 2 bits.

Top	Top-k bits	Bits	WinoGrande (%)	COPA (%)	OBQA (%)	HellaSwag (%)	PIQA (%)	MMLU (%)	Average (%)
1	4	2.29	66.30	83.00	39.00	69.28	75.03	35.02	61.27
	8	2.35	66.14	87.00	39.80	69.44	75.30	34.04	61.95
2	4	2.32	66.38	88.00	38.60	69.44	76.06	36.49	62.49
	8	2.44	65.98	90.00	38.60	69.77	76.33	35.82	62.75
5	4	2.41	66.54	87.00	38.40	70.13	76.12	38.02	62.70
	8	2.70	64.96	89.00	39.40	70.56	75.90	38.56	63.06
10	4	2.55	67.17	86.00	39.20	70.55	76.55	39.11	63.10
	8	3.14	66.06	88.00	39.00	70.81	76.71	39.30	63.31
15	4	2.70	67.17	83.00	39.00	71.72	76.93	40.41	63.04
	8	3.58	65.75	85.00	41.00	71.34	76.39	40.48	63.33
20	4	2.85	67.88	84.00	40.20	72.35	77.69	41.25	63.90
	8	4.02	66.61	89.00	38.00	72.58	77.64	41.25	64.18
25	4	2.99	67.17	87.00	40.00	73.26	78.07	42.38	64.65
	8	4.46	68.67	86.00	41.00	73.00	78.67	41.79	64.86
30	4	3.14	69.69	89.00	40.60	73.92	77.53	42.82	65.59
	8	4.90	67.56	88.00	40.80	73.88	78.56	41.94	65.12

Table 9: Baseline results of the 16-bit (FP16), 4-bit, and 2-bit quantization.

Bits	WinoGrande (%)	COPA (%)	OBQA (%)	HellaSwag (%)	PIQA (%)	MMLU (%)	Average (%)
				Mixtral-8x7B			
16	76.48	93.00	47.00	83.98	82.37	70.35	75.33
4	74.98	92.00	46.20	81.65	80.85	67.65	73.89
2	49.33	63.00	25.40	28.18	52.99	24.29	40.53
				DeepSeek-MoE-16B-base			
16	70.40	91.00	44.20	77.35	78.72	44.77	67.74
4	71.35	87.00	43.20	76.39	78.51	44.22	66.78
2	53.28	76.00	30.20	45.33	66.54	25.28	49.44