MC#: Mixture Compressor for Mixture-of-Experts Large Models

Wei Huang, Yue Liao, Yukang Chen, Jianhui Liu, Haoru Tan, Si Liu *Senior Member, IEEE*, Shiming Zhang *Member, IEEE*, Shuicheng Yan *Fellow, IEEE*, Xiaojuan Qi *Senior Member, IEEE*

Abstract—Mixture-of-Experts (MoE) has emerged as an effective and efficient scaling mechanism for large language models (LLMs) and vision-language models (VLMs). By expanding a single feed-forward network into multiple expert branches, MoE increases model capacity while maintaining efficiency through sparse activation. However, despite this sparsity, the need to preload all experts into memory and activate multiple experts per input introduces significant computational and memory overhead. The expert module becomes the dominant contributor to model size and inference cost, posing a major challenge for deployment. To address this, we propose MC# (Mixture-Compressor-sharp), a unified framework that combines static quantization and dynamic expert pruning by leveraging the significance of both experts and tokens to achieve aggressive compression of MoE-LLMs/VLMs. To reduce storage and loading overhead, we introduce Pre-Loading Mixed-Precision Quantization (PMQ), which formulates adaptive bit allocation as a linear programming problem. The objective function jointly considers expert importance and quantization error, producing a Pareto-optimal trade-off between model size and performance. To reduce runtime computation, we further introduce Online Top-any Pruning (OTP), which models expert activation per token as a learnable distribution via Gumbel-Softmax sampling. During inference, OTP dynamically selects a subset of experts for each token, allowing fine-grained control over activation. By combining PMQ's static bit-width optimization with OTP's dynamic routing, MC# achieves extreme compression with minimal accuracy degradation. On DeepSeek-VL2, MC# achieves a 6.2× weight reduction at an average of 2.57 bits, with only a 1.7% drop across five multimodal benchmarks compared to the 16-bit baseline. Moreover, OTP further reduces expert activation by 20% with less than 1% performance loss, demonstrating strong potential for efficient deployment of MoE-based models.

Index Terms—Mixture-of-Expert, Multimodal Large Language Model, Model Compression, Quantization, Pruning

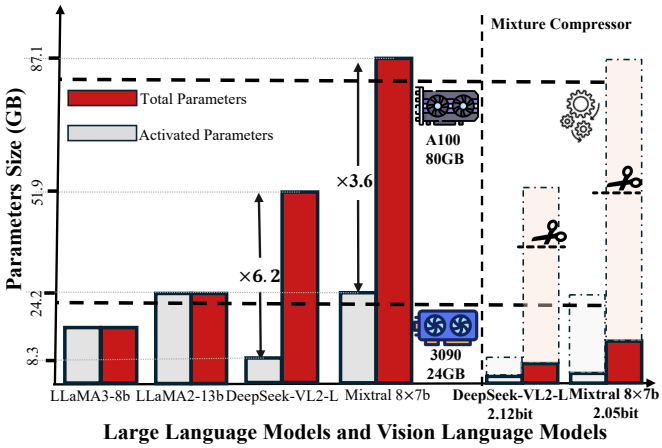

Fig. 1. Comparison of total parameter size and inference activated parameter size on a few open-source large vision/language models and compressed Mixtral 8×7b (MoE-LLMs) and DeepSeek-VL2-L (MoE-VLMs). L: large.

1 Introduction

Mixture-of-Experts large-language/vision-language models (MoE-LLMs/VLMs) [2]–[5] provide an efficient mechanism for model scaling by leveraging sparse architectures, where the router activates only a subset of experts. This selective activation improves computational efficiency and scalability by dynamically assigning experts based on the specific requirements of each input. However, while the number of active experts is reduced to enhance inference efficiency, all experts must still be loaded into memory simultaneously, and multiple experts are typically activated during inference. This results in substantial memory and computational overhead. For instance, a typical MoE-LLM like Mixtral 8×7b [3] requires nearly 90GB of GPU memory (Fig. 1), while MoE-VLMs such as DeepSeek-VL2-L demand over 50GB of GPU memory (Fig. 1), largely due to the heavy computational and memory requirements of their LLM backbone. These high resource requirements hinder the deployment of MoE-based large models on hardware with limited capacity, further driving the need for research into compression techniques for MoE large models.

The primary objective of compressing MoE-based large models is to reduce the size of expert parameters, as they significantly impact memory usage [1], [6]. For example, in models such as DeepSeek-VL2-L, the number of expert parameters is about 77 times larger than that of attention modules. On the other hand, recent studies [1], [7], [8]

- *Wei Huang, Jianhui Liu, Haoru Tan, Shiming Zhang and Xiaojuan Qi are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, Hong Kong SAR. Email: weih@connect.hku.hk, jhliu0212@gmail.com, hrtan@eee.hku.hk, beszhang@hku.hk, xjqi@hku.hk*
- *Yue Liao and Shuicheng Yan are with the School of Computing, National University of Singapore, Singapore. Email: liaoyue.ai@gmail.com, yansc@nus.edu.sg*
- *Yukang Chen is with NVIDIA Research, USA. Email: chenyukang2020@gmail.com*
- *Si Liu is with the Institute of Artificial Intelligence, Beihang University, Beijing, China, and also with Hangzhou Innovation Institute, Beihang University, Hangzhou, China. Email: liusi@buaa.edu.cn.*

- *A preliminary version of this research has appeared in ICLR 2025 [1].*

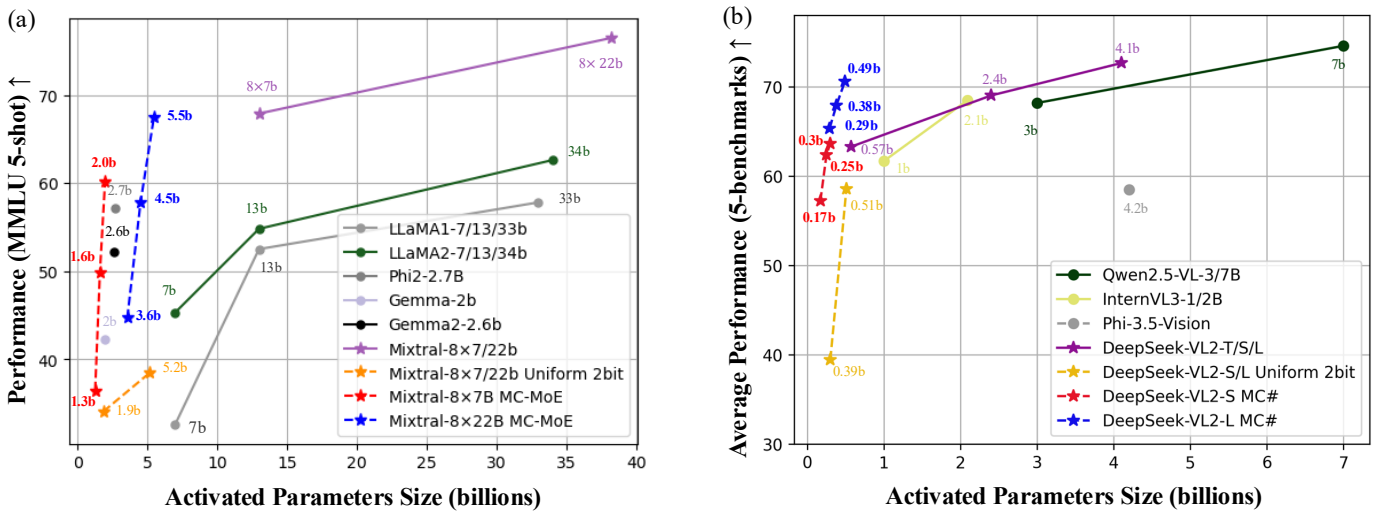

Fig. 2. (a) MMLU (5-shot↑) accuracy across different open-source LLMs with various activated parameters (dot-lines denote the quantized models, solid-lines are 16-bit models). To align quantized models' parameter size with 16-bit models, we define 16bits as one standard parameter (e.g., 8×2-bit elements represent one parameter). (b) Average performance(↑) on 5 general multimodal benchmarks across different open-source VLMs with various activated parameters. L: large, S: small. T: tiny.

have revealed that, due to the training strategies of MoE models, not all experts are equally important, and the importance of input tokens also varies. This suggests that compressing MoE models can benefit from quantizing expert weights during the preloading phase [6] or pruning experts during inference [8]–[10]. However, traditional uniform bit-width quantization struggles to maintain performance under extreme compression ratios, and rule-based expert pruning is often unsuitable for MoE models with a large number of experts, such as DeepSeek-VL2. To address these challenges, this study explores extreme hybrid compression for multimodal MoE models. By introducing expert importance metrics, we apply mixed-precision quantization to statically compress experts. Additionally, we propose an innovative differentiable mask based on Gumbel-Softmax sampling, enabling dynamic top-any expert pruning for different tokens. This approach achieves highly lightweight MoE-VLMs while maintaining an optimal Pareto curve between compression and performance.

We introduce **MC#**, *i.e.*, **M**ixture-**C**ompressor-sharp, a novel framework for multimodal MoE models that integrates expert quantization and pruning for efficient compression. **MC#** operates in two key phases: *Pre-Loading Mixed-Precision Quantization (PMQ)* and *Online Top-any Pruning (OTP)*, as illustrated in Fig. 3.

In the pre-loading phase, we achieve extreme compression of expert parameters through mixed-precision quantization under ultra-low bits. Analysis of MoE-LLMs reveals significant imbalances in activation reconstruction errors, routing weights, and expert activation frequencies [1], with these discrepancies becoming more pronounced in MoE-VLMs (see Section 3.2.1 and Fig. 4). These imbalances motivate assigning distinct bit-widths to different experts in VLMs. We propose a weighted evaluation function that jointly accounts for activation frequency, activation weight, and quantization losses across bit-widths. This function is optimized using a Linear Programming (LP) model to determine the optimal quantization configuration. Utilizing a training-free Post-Training Quantization (PTQ) approach, such as GPTQ [11], our PMQ framework delivers high-performance compression at any-precision in low bit-widths (1.5-bit to 2.5-bit) while striking a Pareto-optimal balance between model size and performance. Furthermore, our mixed-precision strategy seamlessly integrates with state-of-the-art quantization techniques [12]–[16]. As for the inference phase, to address the inflexibility of previous rule-based expert pruning methods [1], [8], we propose *Online Top-any Pruning (OTP)*, a learnable dynamic pruning approach that operates in an end-to-end training mode with only a small batch of data. In the *top-k* MoE mechanism, dynamic pruning of experts requires selective activation based on the varying importance of tokens within a limited set of experts. However, the non-differentiability of mask selection prevents direct backpropagation for learning. To overcome this limitation, we model mask selection as a probabilistic sampling process, transforming it into a stochastic operation. By leveraging Gumbel-Softmax [17], we make the sampling process differentiable, enabling the optimization of each mask candidate's probability through gradient descent. During training, OTP aims to learn an appropriate token-wise expert mask distribution, further reducing the number of activated experts and single-device computational costs in ultra-low-bit MoE models, all while maintaining the original performance.

Compared to uniform quantization or other mixed-precision strategies, the proposed **MC#** approach significantly enhances the performance of compressed MoE-LLMs/VLMs. Specifically, when compressing DeepSeek-VL2-L to approximately 2.57 bits, its activation parameters are reduced to just 0.49b, while maintaining only 1.7% performance loss across five common multimodal benchmarks. Remarkably, under the same activation parameter budget, it even outperforms uncompressed state-of-the-art small-scale models, as shown in Fig. 2(b). This mixture compression strategy fully exploits the unique characteristics of experts, enabling high performance even under extreme compression

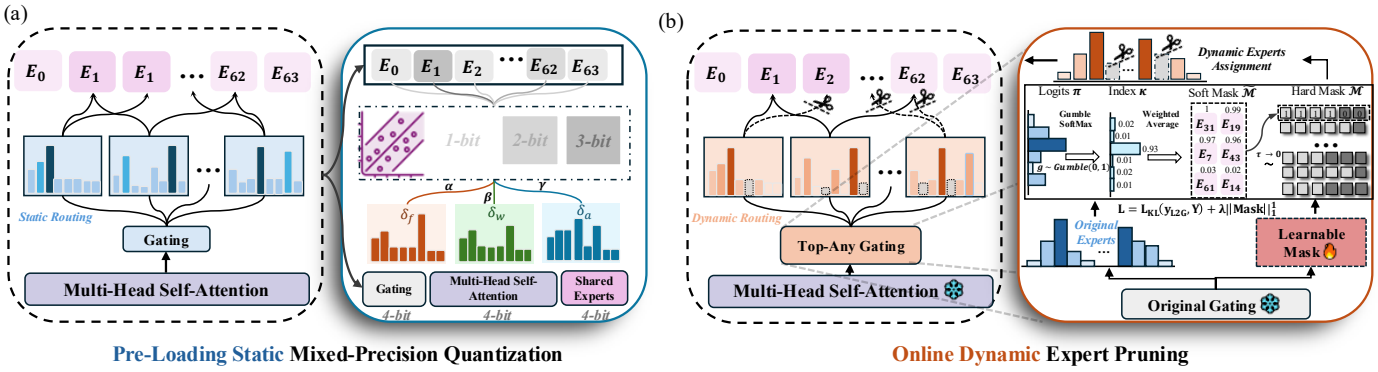

Fig. 3. The overview of our proposed MC pipeline with two-stage compression for experts. (a) Framework of pre-loading static mixed-precision quantization (PMQ) of MoE-LLMs. PMQ determines the activated feature and loss sensitivity of all experts and plans the optimal precision configuration under ultra-low-bit-width. (b) Schematic of online top-any pruning (OTP) of MoE-LLMs. OTP utilizes a learnable experts pruning scheme to achieve higher inference efficiency.

conditions. These results highlight the immense potential and practicality of compressing multimodal MoE models, making them scalable from multi-GPU cluster environments to consumer-grade and edge-level applications.

This paper presents substantial extensions to the conference version [1] in the following aspects. (i) We conduct an in-depth analysis of the uneven expert activation patterns and compression challenges in MoE-VLMs, such as DeepSeek-VL2, providing valuable insights for effectively deploying mixed-precision quantization in multimodal MoE models. (ii) We propose a differentiable online dynamic expert mask learner based on Gumbel Softmax, which leverages a soft mask mechanism to learn an optimal token-wise expert pruning strategy even with a small number of samples. Additionally, we introduce expert sparsity control constraints to explore the optimal pruning strategies under specific pruning ratios. Our proposed *OTP* strategy successfully overcomes the limitations of rule-based expert pruning methods, demonstrating robust scalability to scenarios with a larger number of experts. (iii) We comprehensively evaluate the compression performance of **MC#** on nine language benchmarks and six multimodal large model benchmarks. By combining static and dynamic compression, our approach achieves the state-of-the-art compression performance than other methods and previous MC [1]. (iv) We have demonstrated that our proposed training-free multi-factor mixed-precision quantization method achieves the Pareto-optimal boundary in the ultra-low bit-width range across various types of large MoE models. Furthermore, when combined with a learnable expert pruning strategy, our approach achieves higher compression ratios with significantly lower performance loss. In summary, these advancements contribute to building a more efficient and unified mixed compression framework for MoE-based LLMs and VLMs, showcasing significant potential and applicability across various real-world scenarios.

2 Related Works

2.1 Mixture-of-Experts Large Models

Large-scale models built on LLM-based architectures have achieved remarkable progress in various natural language processing [18], [19] and multimodal understanding domains [4], [5], [20]. Sparse-activation MoE models have emerged as a key strategy for improving the trade-off between performance and efficiency in LLMs and VLMs. In MoE models, each layer consists of multiple expert submodels, and only a specific subset of experts is activated for each token. This mechanism significantly enhances efficiency compared to dense models that activate all parameters for every input [21], [22], while also delivering stronger representational power and improved multimodal understanding. Recent advancements in LLMs [23] have further extended the capabilities of MoE-based architectures. Industry-leading models, such as DeepSeek-VL2 [4], leverage MoE-LLM backbones to achieve state-of-the-art performance. Despite their success, these models bring more memory requirements on GPU storage, presenting new challenges for efficient deployment [24], [25].

2.2 Quantization for LLMs

Post-Training Quantization (PTQ) is an efficient approach that requires no additional training, making it particularly suitable for LLMs, such as GPTQ [11], AWQ [26], and MBQ [27], and other activation quantization works [14], [28]. Previous studies [29]–[31] have explored the significance of weight diversity and proposed mixed-precision strategies to improve low-bit performance by assigning different bit-widths accordingly. Examples include unstructured relaxed precision methods like SpQR [32] and structured mixed-precision approaches such as SliM-LLM [33]. Recent work has introduced expert-guided block-wise quantization benchmarks for MoE-LLMs to address disparities in expert weights [6]; however, significant performance degradation persists under ultra-low-bit quantization. Codebook-based encoding methods offer more precise quantization for LLMs and can further enhance post-quantization performance through fine-tuning [12], [15]. Although Quantization-Aware Training (QAT) demands substantial resources [13], [34], QAT-based retraining strategies, or PTQ combined with additional fine-tuning [16], [35], [36], have proven to be more effective in preserving the performance of lightweight, quantized models. Since the LLM component in MoE-VLMs contains the largest number of parameters and incurs the highest

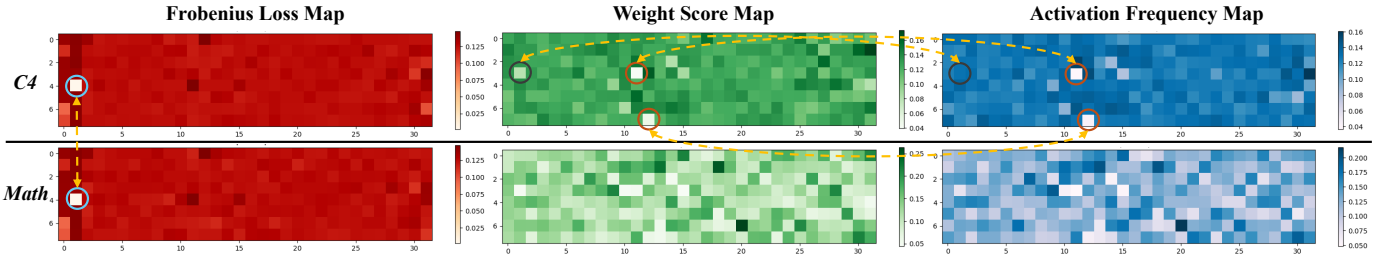

Fig. 4. Distribution of expert drop F-norm (red), activated weights (green) and frequencies (blue) in the Mixtral 8×7b model, encompassing 32 MoE layers with 8 experts per layer. The top set of the heatmap is calculated through C4 dataset [43], and the bottom set is calculated through MATH dataset. MoE-LLMs selectively activate top-2 experts in each MoE layer, wherein a significant portion of experts remain less important or inactivated all the time.

inference cost, these PTQ techniques can also be effectively adapted for use in VLMs [36].

2.3 Parameter pruning for LLMs

Parameter pruning is another effective neural network compression method [37], [38], and it has recently become crucial for reducing the size of LLM weights [39]–[41]. Traditional pruning techniques typically focus on two approaches: structured pruning and unstructured pruning, both of which selectively zero out parameters based on their importance [24]. In the context of large MoE models, less important expert models can be pruned based on activation frequency or gating statistics [9], [10], [42]. During the model pre-loading stage, pruning often incurs greater performance loss than quantization at the same compression ratio. However, while dynamically adjusting quantization bit-widths during inference remains challenging, pruning offers the flexibility to dynamically select active parameters during inference [25]. Recent studies, such as [8], have explored dynamic activation of *top-k* (where $k = 1$ or 2) experts in MoE models based on gating weights, significantly improving inference efficiency. However, such straightforward rule-based dynamic activation strategies lack robustness when scaling to larger k, as in models like DeepSeek-VL2 ($k = 6$) [4].

3 Mixture Compressor

3.1 Preliminaries

Mixture-of-Experts VLMs. In decoder-only MoE models, conventional feed-forward networks (FFN) are replaced by MoE layer, each having N experts [45]. The MoE model selectively activates the *top-k* experts for different tokens by a group of routing scores $\mathbf{w}_{top\text{-}k} = \{w_0, w_1, ..., w_{k-1}\}$, $k \leq N$, generated by a gating layer $\mathrm{G}(\mathbf{t})$. Fig. 3(a) also illustrates the experts selection mechanism during the inference phase based on routing scores. For instance, in the DeepSeek-VL2 model, there are 72 dynamically activated experts and two shared experts. The i^{th} token in input tokens $T \in \mathbb{R}^{L \times D}$ is routed to the *top*-6 experts [4]:

$$\mathbf{y} = \sum_{w_j \in Top\text{-}6\{\mathrm{G}(\mathbf{t_i})\}} w_j \, \mathrm{F}_j(\mathbf{t_i}) + \mathrm{F}_s(\mathbf{t_i}), \tag{1}$$

where F_j represents the feed-forward operator of the $j\text{-}th$ selected expert, F_s represents the feed-forward operator of the shared experts and w_j denotes the $j\text{-}th$ routing weights calculated by the gating $\mathrm{G}(\mathbf{t_i})$. Therefore, according to the definition in Eq. 1, the routing mechanism establishes the correspondence between tokens and experts.

Quantization Technique. Quantization is regarded as an effective method to compress the model weights. Specifically, floating-point weights distributed in the interval $[\mathbf{W}_{min}, \mathbf{W}_{max}]$ are mapped to the integer range of $[0, 1..., 2^b]$, where b represents the target bit-width, and the quantization reconstruction for the weights $\mathbf{W} \in R^{n \times m}$ can be defined as:

$$\arg\min_{\mathbf{W}^q} \quad \|\mathbf{W}\mathbf{X} - \mathrm{Q}(\mathbf{W})\mathbf{X}\|_2^2, \tag{2}$$

where $\mathrm{Q}(\cdot)$ denotes the quantization function and $\|\cdot\|_2$ is the mean square error (MSE) loss. $\mathrm{Q}(\cdot)$ is generally designed as:

$$\begin{cases} \hat{\mathbf{W}}_q = \mathrm{clamp}(\lfloor \frac{\mathbf{W}}{s} \rceil + z, 0, 2^N - 1), \\ s = \frac{\mathbf{W}_{\max} - \mathbf{W}_{\min}}{2^b - 1}, z = -\lfloor \frac{\mathbf{W}_{\min}}{s} \rceil \end{cases} \tag{3}$$

where $\hat{\mathbf{W}}_q$ indicates quantized weight which is integer, $\lfloor \cdot \rceil$ is round operation and $\mathrm{clamp}(\cdot)$ constrains the value within integer range (e.g. $[0, 1, 2, 3]$, $b = 2$). Δ is scale factor and z is quantization zero point, respectively. In 1-bit condition, weights are further quantized with binary values (-1 or $+1$):

$$\begin{aligned} &\hat{\mathbf{W}}_b = \mathrm{sign}(\mathbf{W}), \alpha = \frac{||\mathbf{W}||_{\ell 1}}{m} \\ &\mathrm{sign}(\mathbf{W}) = \begin{cases} 1 & \text{if } w \geq 0, w \in \mathbf{W}, \\ -1 & \text{others.} \end{cases} \end{aligned} \tag{4}$$

where $\hat{\mathbf{W}}_b$ is binary result. α denotes binarization scales [31] in channel-wise manner [46]. The primary objective of this study is to explore the optimal mixture compression strategy for MoE-VLMs. To this end, we employ the efficient PTQ scheme, GPTQ [11], as our foundational tool. By utilizing Hessian-based estimation ($\mathbf{H} = 2\mathbf{X}\mathbf{X}^\top$) and quantization error compensation, GPTQ effectively reduces the group-wise quantization error of weights, enabling the quantization of DeepSeek-VL2-L within 30 minutes. Our static quantization method is orthogonal to other quantization techniques. Current PTQ methods [14], [26], codebook-based works [12], [15], and even the deployment of fine-tuning [16] or QAT [13], [34] can be deployed for **MC#**.

Rule-based Experts Pruning. To effectively perform dynamic experts pruning, an intuitive and efficient method involves utilizing the *top-k* experts' routing scores during inference [8],

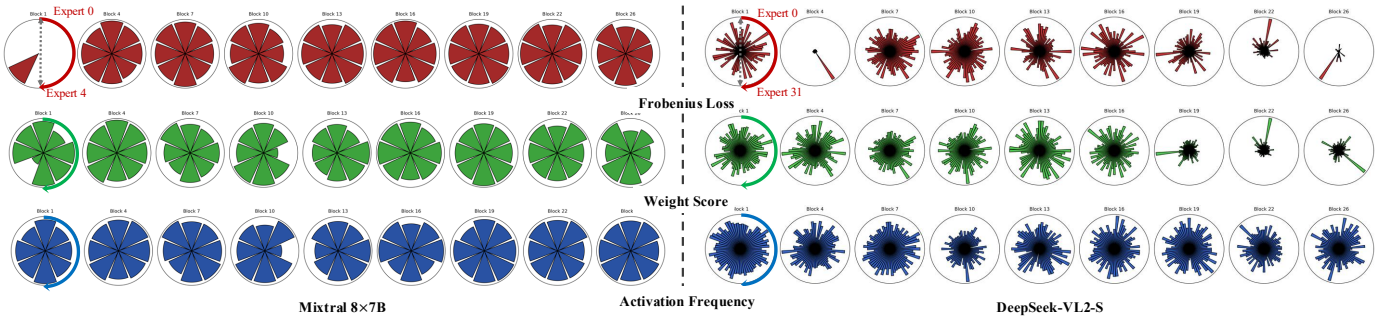

Fig. 5. Comparison of experts quantization loss and activations between MoE-LLM and MoE-VLM. The left panel illustrates the quantization loss and the distribution of expert activation features for Mixtral 8×7b calibrated on a subset of C4 dataset [43], while the right panel presents the corresponding metrics for DeepSeek-VL2-S calibrated on a subset of the M4 dataset [44]. The expert indices are arranged in a clockwise manner, covering experts 0-8 and 0-64, respectively. Notably, the quantization loss and activation feature distributions across different experts in MoE-VLMs are significantly more imbalanced compared to those in MoE-LLMs.

[47]. This approach directly skips experts with lower routing weights among the selected set for each token. For simplicity, when $k = 2$ (as in Mixtral 8×7b), the pruning process follows:

$$\{w_0 = 0, w_1 = 1, w_0, w_1 \in Top\text{-}2\{\mathrm{G}(\mathbf{t})\} \mid \frac{w_1}{w_0} < \mu\}, \quad (5)$$

where, w_0 and w_1 denote the gating weights of top-2 experts, respectively, with μ erving as a hyperparameter threshold for each MoE layer. This threshold is set at the median value of $\frac{w_1}{w_0}$ derived from calibration data [8]. According to Eq. 5, when a selected expert has a significantly lower weight, it can be pruned from the current set of candidates, leaving only the primary experts for computation. However, in models like DeepSeek-VL2, which feature a larger number of experts and a greater value of k, rule-based approaches struggle to handle the diversity of combinations effectively.

3.2 Pre-Loading Mixed-precision Quantization

As discussed in Sec.3.1, the primary storage overhead in MoE-VLMs lies within the experts, necessitating effective compression before deployment on devices. While mainstream pruning methods often suffer from significant performance degradation under extreme pruning conditions (e.g., $\geq$50%) [8], [39], [40], quantization has proven to achieve substantial compression with comparatively lower performance loss [24], [36]. However, as highlighted in [6] and our findings in Sec. 4.2, uniform bit-width quantization fails to meet the stringent accuracy requirements under extreme compression scenarios for MoE-VLMs. This limitation, coupled with the diverse and uneven characteristics of experts, motivates us to investigate optimal mixed-precision quantization strategies.

This section presents our *Pre-Loading Mixed-Precision Quantization (PMQ)* method, which aims to efficiently minimize model size through targeted expert quantization. The central objective of PMQ is to optimize the bit-width allocation strategy for experts. To achieve this, we first perform a comprehensive analysis of expert behavior on the calibration dataset, using these insights to construct an Integer Programming (IP) model that determines the optimal quantization configuration. Meanwhile, for other model components, such as attention parameters, a uniform bit-width is applied across the board.

3.2.1 Experts Significance Analysis

The core principle of our expert quantization strategy is to allocate bit-widths based on the relative importance of each expert within a MoE block. Initially, we analyzed the performance of various experts in the Mixtral 8×7b configuration, including their reconstruction loss (Frobenius norm) [48], as well as activation patterns on the C4 dataset [43] and the domain-specific Math dataset [49]. As illustrated in Fig. 4, the impact of experts on the model varies significantly: 1) certain experts, such as the one located at position $[2, 4]$ (Fig. 4, left), exhibit minimal influence on the output activation reconstruction loss, while others in the second layer demonstrate substantially higher loss values, highlighting the imbalance among experts; 2) activation scores and frequencies follow distinct patterns, where experts at positions $[11, 3]$ and $[12, 7]$ show extremely low activation frequencies and average scores, while the expert at $[1, 3]$ has relatively low scores but notably higher activation frequency; 3) in task-specific scenarios, such as mathematical reasoning, fewer experts are activated, resulting in a sparser distribution compared to general tasks. This variability in routing characteristics necessitates considering multiple factors when determining the optimal bit-width allocation for experts. As shown in Fig. 5 (b), we further visualized the expert quantization loss and activation patterns for MoE-VLMs (e.g., DeepSeek-VL2-S) on the M4 dataset [44], where similar trends were observed. Moreover, compared to MoE-LLMs (Fig. 5 (a)), the imbalance is even more pronounced, suggesting that MoE-VLMs are better suited for mixed-precision quantization schemes.

3.2.2 Weighted Importance Factors

In assessing the significance of each expert in MoE-LLMs, we primarily consider two factors: access frequency and activation weight. Using an N-sized calibration dataset, such as C4 (general language understanding) for MoE-LLMs or M4 (visual question answering) for MoE-VLMs, inference is performed on the original 16-bit MoE models. Access frequency, defined as the rate at which an expert is activated, is expressed as $\phi_i = \frac{n_i}{N}$, where n_i represents the total activations of the i-th expert. A higher frequency suggests greater generality and applicability across diverse tokens. However, relying solely on frequency may overlook the importance of less frequently activated experts. To address

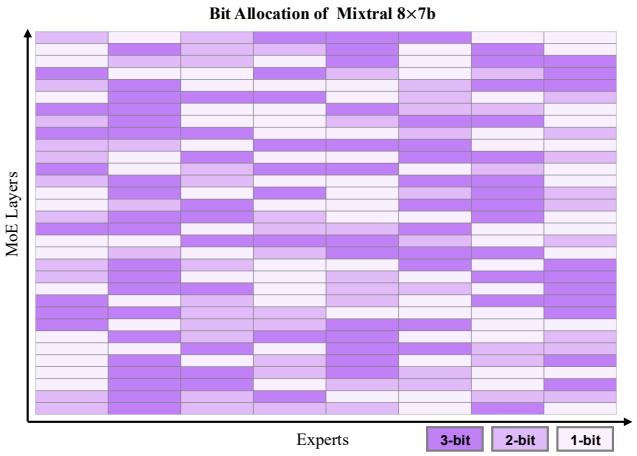

Fig. 6. Visualization on different bit-width allocation of Mixtral 8×7b (MoE-LLMs). Average bit-width is 2-bit in 32 MoE layers, each with 8 experts. Color refers to the bit size.

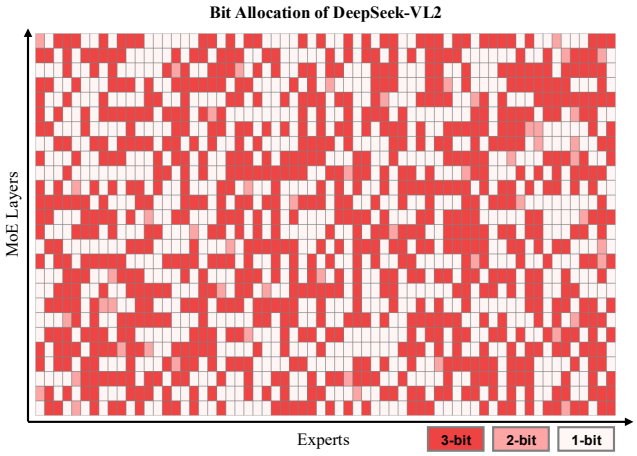

Fig. 7. Visualization on different bit-width allocation of DeepSeek-VL2-S (MoE-VLMs). Average bit-width is 2-bit in 26 MoE layers, each with 64 experts. Color refers to the bit size of each expert.

this limitation, an activation-weighted metric is introduced, calculated as $w_i = \frac{\sum_{j=1}^{N} \sigma_j}{N}$, where σ_j denotes the routing weight assigned to the expert during the j-th inference. This metric provides a more nuanced understanding of each expert's contribution by incorporating the significance of routing weights. The overall importance of an expert is then computed as $\phi_i^{\alpha} \cdot w_i^{\beta}$, where α and β are hyperparameters that balance these two parts.

3.2.3 Optimal Experts Bit-Width Allocation

Building on the determination of expert significance, we explore its application in mixed-precision quantization by assigning varying bit-widths to experts based on their importance. This approach ensures that more significant experts retain higher bit-widths, while less significant experts undergo more aggressive quantization. Beyond leveraging expert significance, we also assess the reconstruction error of output activations in each MoE layer after quantization, enabling a precise evaluation of the impact of quantizing individual experts. Specifically, for each expert, we compute the Frobenius norm (F-norm) between the model's output when the expert is quantized and the output generated when no experts are quantized, providing a quantitative measure of the effect of quantization on model performance:

$$\epsilon_{i,j} = \|\mathscr{F}(\theta) - \mathscr{F}(\theta[e_i \to Q(e_i, j)])\|_F, \tag{6}$$

where $\mathscr{F}(\theta)$ is the model output with full parameters θ, and $\mathscr{F}(\theta[e_i \to Q(e_i, j)])$ represents the output when only expert e_i is quantized to j bits. $Q(\cdot)$ denotes the quantization function.

To achieve an extremely low average bit-width of b across all experts in an MoE block, with bit-width options constrained to $1, 2, 3$ bits, we formulate the task as an Integer Programming (IP) optimization problem. This approach allows us to efficiently compute the optimal bit-width allocation for each expert, ensuring that the targeted average bit-width is met. Remarkably, the IP optimization process is computationally efficient, completing the allocation computation within a single second. The IP model is defined as follows:

$$\begin{aligned} \text{MINIMIZE} \quad & \sum_{i=1}^{n} \sum_{j=1}^{3} \phi_i^{\alpha} \cdot w_i^{\beta} \cdot (\epsilon_{i,j} \cdot x_{ij})^{\gamma} \\ \text{SUBJECT TO} \quad & \sum_{i=1}^{n} \sum_{j=1}^{3} j \cdot x_{ij} = n \cdot k, \quad \sum_{j=1}^{3} x_{ij} = 1, \quad \forall i, \\ & \sum_{i=1}^{n} x_{i3} \geq 1, \quad \sum_{i=1}^{n} x_{i2} \geq 1, \quad x_{ij} \in \{0, 1\}, \quad \forall i, j. \end{aligned} \tag{7}$$

In this framework, x_{ij} is defined as a binary variable indicating whether the i-th expert is quantized to j bits ($x_{ij} = 1$ if true, otherwise $x_{ij} = 0$). To ensure the accuracy of critical experts, we impose a constraint that requires at least one expert to be quantized to 3 bits and another to 2 bits. The parameter γ serves as a weighting hyperparameter to balance the optimization process. Once the optimal bit-width combination for each MoE block expert is determined, the GPTQ quantization algorithm is applied to quantize the experts accordingly. For the remaining weights, such as those in the attention mechanism, gating module, and shared experts, their minimal parameter size and contribution to the output allow us to follow the recommendation from [6] and quantize them uniformly to 4 bits, contributing an additional average bit-width of no more than 0.08 bits. Fig. 6 and Fig. 7 display the precision distributions of different experts.

3.3 Binary Weight Saving and Dequantization

This paper presents **MC#**, which explores static compression strategies and dynamic pruning methods for MoE-LLMs in the ultra-low bit-width range, with selected static bit-width of 1-bit, 2-bit, and 3-bit. We observe that both 2-bit and 3-bit can be addressed using conventional linear quantizers, a method commonly utilized in most studies [11], [14], [26], [33]. We utilize the HQQ [50] tool to save quantized weights and handle dequantization. In contrast, the quantization of 1-bit weights involves totally different calculations; we first provide the binarization formula for the weights: where $\mathbf{W} \in \mathbb{R}^{d \times m}$ is the full precision weight and $\mathbf{B} \in \{-1, +1\}^{d \times m}$ denotes the binarized matrix. Due to the elements' range of $\mathbf{B}$ being

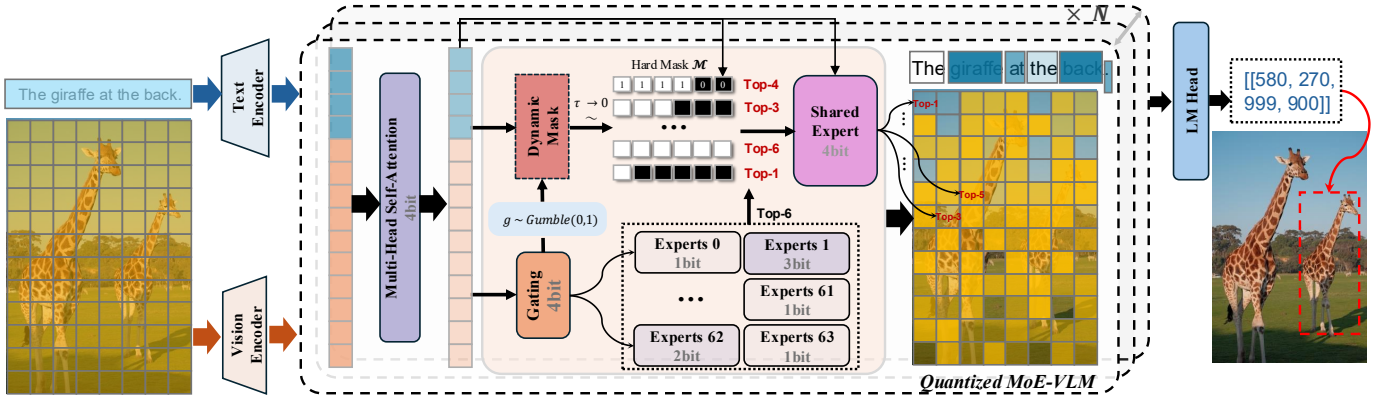

Fig. 8. The workflow of Online Top-any Pruning (OTP). In this framework, each token from multimodal data is input into the quantized DeepSeek-VL2 model, where six experts are selectively activated. A learnable dynamic mask is then applied to dynamically prune redundant experts among the six, retaining only the most relevant ones for computation alongside shared experts. A key feature of OTP lies in its ability to dynamically assess the importance of tokens and flexibly prune experts, ensuring efficient and adaptive processing.

±1, we can not directly save the one-bit value into compact memory. Hence, we propose a simple transformation for $\mathbf{B}$:

$$\widetilde{\mathbf{B}} = \frac{\text{sign}(\mathbf{W}) + 1}{2} \tag{8}$$

where $\widetilde{\mathbf{B}} \in \{0,1\}^{d\times m}$. In this case, we can use a 1-bit memory to store each element. During the inference stage, we need to dequantize the binary weight and perform the matrix multiplication of each input vector as follows:

$$s \cdot \mathbf{xB} = s(\sum_{j:\widetilde{\mathbf{B}}_{ij}=1}^{d} \mathbf{x}_j - \sum_{j:\widetilde{\mathbf{B}}_{ij}=0}^{d} \mathbf{x}_j), \text{for } i = 1, 2, ...m \tag{9}$$

where $\mathbf{x} \in \mathbb{R}^{1\times d}$ denotes one set of input token, and s represents the scaling factor of each binary matrix, which is calculated from $s = \frac{\|\mathbf{W}\|_{\ell_1}}{d\times m}$ [46]. In this binarized weight format, we can achieve computation without minimal multiplication. As shown in Eq. (12), the original computation requires dm multiplications and $(d-1)m$ additions, resulting in a Multiply-Accumulate Operations (MACs) consumption of dm and a computational complexity of $O(m^2)$. In contrast, binary matrix operations require only m multiplications and $(d-1)m$ additions, leading to a MACs consumption of just m and a computational complexity of $O(m)$.

3.4 Online Top-any Pruning

To address the challenges of rule-based expert pruning, we propose and introduce a learnable dynamic expert selection framework, *Online Top-any Pruning (OTP)*, which can be applied to both MoE-LLMs and MoE-VLMs with any-experts. This framework further enhances inference efficiency on lightweight models. We define the pruning of *top-k* selected experts as a mask selection problem, with the candidate set size denoted as $|C|$. Inspired by rule-based exploration, we prioritize masking gating weights with lower values, thus setting the size of $|C|$ to $|C|$. Given a candidate group of $|C|$ expert weights, denoted as $E \in \mathbb{R}^{1\times k}$, the objective of dynamic pruning is to identify the optimal binary mask $M^* \in \mathbb{B}^{1\times k}$. Specifically, in the case of DeepSeek-VL2 with $k = 6$, the discrete candidate set C is defined as:

$$\begin{aligned} C_k &= \{M \in \mathbb{B}^{1\times k} \mid 1 \le \sum M \le 6\} \\ &= \{\hat{M}_0, \hat{M}_1, \hat{M}_2, \hat{M}_3, \hat{M}_4, \hat{M}_5\} \\ &= \{[1,1,1,1,1,1], [1,1,1,1,1,0], [1,1,1,1,0,0], \\ &\quad [1,1,1,0,0,0], [1,1,0,0,0,0], [1,0,0,0,0,0]\}, \end{aligned} \tag{10}$$

In MoE-VLMs, we first rank the *top-k* experts based on their weights, assigning a value of 0 to denote the pruned experts, as shown in Fig. 8. To accurately prune unnecessary tokens while maintaining the model's performance, we define the objective function for the learnable mask in each MoE layer as follows:

$$\arg\min_{M_{i,j}\in C_k} \mathcal{L}_D(\mathbf{X}; \{\mathrm{G}(\mathbf{t_i})_k \odot M_{i,j}\}), \tag{11}$$

where $\mathrm{G}(\mathbf{t_{i,j}})_k$ is the *top-k* index activated by original gating, $\mathcal{L}_D$ represents the distillation loss associated with the final output logits with non-masked MoE models, and the operator $\odot$ denotes element-wise multiplication used to prune the experts. $M_{i,j}$ is defined as the mask results of the i^{th} token in the j^{th} MoE layer. However, due to the non-differentiable nature of mask selection, we proceed by reformulating the mask selection process into a sampling mechanism. This transformation enables the model to learn the relationship between input tokens and the mask through optimization, making the process trainable and more adaptable.

3.4.1 *Differentiable Dynamic Experts Pruning*

To effectively model the mask sampling operation, we employ the Gumbel-Softmax [17], [41] technique to approximate the target mask matrix M_i^*. This approach leverages reparameterization to decompose the stochasticity of sampling into a noise variable. Specifically, it introduces a method for drawing samples from a categorical distribution p. The technique generates a one-hot index y for sampling, thus enabling a differentiable approximation of discrete sampling operations.

$$y = \text{onehot}(\arg\max[log(p_i) - log(-log\delta_i)]), \delta_i \sim U(0,1), \tag{12}$$

TABLE 1
Parameter configuration of learnable router.

	Model	FC1	FC2	Mask
LLMs	Mixtral 8×7b	4096×2	4×2	2×2
	Mixtral 8×22b	6144×2	4×2	2×2
VLMs	DeepSeek-VL2-L	2569×6	12×6	6×6
	DeepSeek-VL2-S	2048×6	12×6	6×6
	DeepSeek-VL2-T	1792×6	12×6	6×6

where $-log(-log(\delta_i))$ refers the Gumbel noise and δ_i is a random noise in uniform distribution. We then design a learnable router, denoted as $DM(\cdot)$ (only two linear layers of each MoE block, as shown in Tab. 1) to generate the categorical distribution p. To solve the issue of differentiable sampling in onehot and $\arg\max$ operation, the approximation of the index is further designed as:

$$\hat{y}_i = \frac{\exp((\mathrm{DM}(\mathbf{t_i}, w) - log(-log\delta_i))/\tau)}{\sum_j \exp(\mathrm{DM}(\mathbf{t_i}, w) - log(-log\delta_i))/\tau)} \tag{13}$$

where the input of the learnable router is the token t_i and its original gating weights w. τ refers to the temperature parameters, adjusting the sharpness of sampled results. As $t \to 0$, the predicted value $\hat{y}_i$ will asymptotically approach the one-hot format described in Eq. 12. Then, we can get $M_i = \hat{y}_i \times C_k$. This operation generates a candidate expert mask based on soft indexing. As illustrated in Fig. 3(b), all operations are differentiable, making it possible to efficiently and elegantly learn an expert mask matrix by introducing a learnable router with a negligible number of additional parameters for each token in different layers.

3.4.2 Learning Objective for MoE Model

During the training process of the learnable mask router, we take into account both the distribution of input tokens and the expert weights derived from the original gating equation, thereby modeling a token-aware dynamic expert pruning mechanism. This approach allows the optimal distribution to be learned in an end-to-end manner. However, when solely optimizing the loss defined in Eq. 11, the model tends to converge towards learning an all-ones matrix. While the loss term $\mathcal{L}_D$ steadily decreases, the model exhibits a tendency to avoid masking any experts. To encourage the model to learn an appropriate dynamic expert pruning strategy under sparsity constraints, we introduce an additional sparsity regularization term:

$$\mathcal{L} = \mathcal{L}_D(\mathbf{X}; \{\mathrm{G}(\mathbf{t_i})_k \odot M_{i,j}\}) + \lambda \|\{M\}\|_1 \tag{14}$$

The sparsity constraint is applied to the ℓ_1 norm of the mask learned within the training batch, where λ serves as a hyperparameter to balance the influence of the sparsity constraint.

4 Experiments

In this section, a series of experiments are conducted to evaluate the proposed **MC#**. We present by describing the experimental parameter settings and results. In Sec. 4.2, we assess the parameter settings for the PMQ method and the performance of mixture quantization. We conduct a detailed evaluation of the performance loss and compression efficiency of OTP stage, shown in Sec. 4.3. Finally, we present the combined performance of MoE mixture compressor.

4.1 Experimental Setup

We implemented hybrid compression deployment for both MoE-LLMs and MoE-VLMs. For the MoE-LLMs, we selected the classic open-source models Mixtral 8×7b and Mixtral 8×22b as our targets, while for the MoE-VLMs, we adopted the DeepSeek-VL2 series, including DeepSeek-VL2-L/S/T (large/small/tiny), as outlined in Tab. 3. The Mixtral 8×7b model can be compressed on two NVIDIA A100-80GB GPUs, whereas the Mixtral 8×22b model requires four NVIDIA A100-80GB GPUs. All models in the DeepSeek-VL2 series can be compressed on a single NVIDIA A100-80GB GPU. During the PMQ phase, the mixed-precision scaling factors were calibrated using the C4 dataset [43], consisting of 128 groups of randomly sampled sequences, each containing 2048 tokens. After determining the bit-width configuration, the final quantization process followed the GPTQ procedure [11]. Additionally, in the dynamic pruning router learning for OTP, we trained using 4096 randomly sampled data (C4 for LLMs and M4 [44] for VLMs).

Building on conclusions from prior studies [6], the weights of components such as attention and shared experts were set to 4-bit precision. Since expert weights constitute the majority of parameters, quantizing other parameters to 4-bit has a negligible impact on the average bit-width, thereby maintaining an ultra-low overall bit-width. In the performance experiments conducted for the proposed **MC#**, perplexity (PPL↓) was chosen as the primary metric to evaluate token prediction capabilities for MoE-LLMs, using the general text dataset WikiText2. To comprehensively assess the compressed language capabilities, the overall performance of the model was further evaluated across eight zero-shot benchmarks (↑) from the EleutherAI LM Harness [51]. For MoE-VLMs, six representative multimodal benchmarks (↑) were selected, aligned with the comparison benchmarks used in DeepSeek-VL2. To ensure consistency, all VLM evaluations were conducted using the VLMEvalKit framework [52].

4.2 Experiment on Pre-Loading Mixed-Precision Quantization

4.2.1 Ablation of Bit-Width Allocating Metrics

To demonstrate the advantages of our PMQ method, we compare the performance of the PMQ quantization curve with results obtained using only gating weights, gating frequencies, Frobenius norm, Hessian-based method, and uniform 2-bit quantization. Fig. 9 and Fig. 10 illustrate this comparison by presenting the PPL performance on Mixtral 8×7b and the average results across five general benchmarks for DeepSeek-VL2-S, respectively. While using expert routing scores derived solely from calibration data shows an improvement over random assignment, the PPL curve remains relatively high. However, activation frequency outperforms weights in achieving better performance.

Moreover, in traditional neural networks and dense LLMs, Hessian-based bit-width allocation has been a commonly

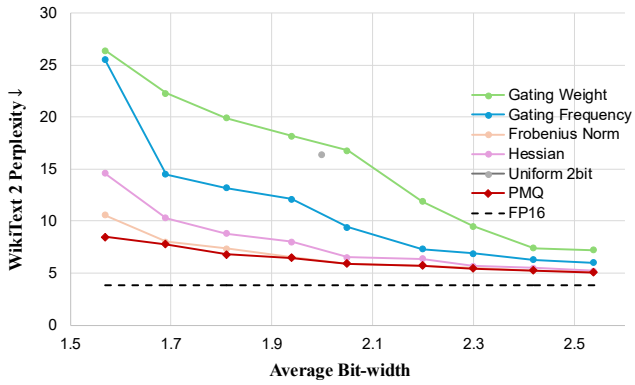

Fig. 9. Quantized performance of Mixtral 8×7b under different mixed-precision strategies (WikiText 2 ↓).

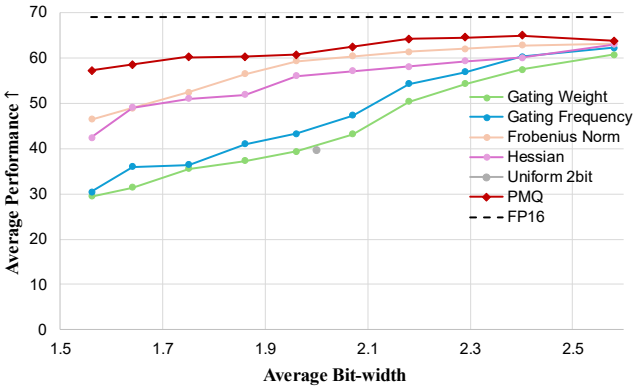

Fig. 10. Quantized performance of DeepSeek-VL2-S under different mixed-precision strategies (MMBench, MMStar, MMMU, AI2D, OCR-Bench ↑).

TABLE 2
Performance of quantized Mixtral 8×7b on eight zero-shot benchmarks. We deploy GPTQ as our baseline PTQ method for uniform quantization. "Uni" denotes the uniform quantization of 2-bit with GPTQ. Since the results of some data sets in the block score predictor (BSP) [6] method were not reported, we resumed the relevant quantized model from the official code repository and evaluated all the results under the same settings. In BSP, 25% MoE layers are 4-bit and the left are 2-bit to achieve 2.54-bit. "HellaS." is the short format of "HellaSwag" and "Wino." denotes "Winogrande". ↓ gives the accuracy loss between quantized results and original 16-bit model.

Method	Bits	PIQA	ARC-e	ARC-c	BoolQ	HellaS.	Wino.	MathQA	MMLU	Avg.% ↑
	16.00	85.20	84.01	57.17	85.35	81.48	75.93	39.29	67.88	71.29
Uni	3.00	82.10	78.58	55.80	82.94	79.28	74.19	39.26	60.58	69.09$_{2.2\%\downarrow}$
Uni	2.00	61.98	47.20	25.71	62.39	41.91	53.22	22.79	30.36	42.67$_{28.6\%\downarrow}$
BSP [6]	2.54	68.23	54.97	28.38	68.16	55.61	62.19	24.07	27.74	49.07$_{22.2\%\downarrow}$
Hessian [29]	2.54	80.21	76.38	51.20	81.11	78.05	72.97	35.27	56.21	67.18$_{4.1\%\downarrow}$
	2.05	75.32	67.26	45.01	70.29	71.90	69.11	31.07	40.85	58.85$_{12.4\%\downarrow}$
	1.57	65.26	52.12	21.84	68.21	52.91	50.32	24.99	31.58	45.91$_{25.4\%\downarrow}$
PMQ	2.54	**80.52**	**77.10**	**51.28**	**82.54**	**79.03**	**73.95**	**39.18**	**56.37**	**67.50**$_{3.8\%\downarrow}$
	2.42	**80.36**	**75.76**	**50.17**	**80.00**	**78.13**	**73.09**	**34.97**	**53.22**	**65.71**$_{5.6\%\downarrow}$
	2.30	**83.11**	**73.59**	**47.78**	**80.83**	**76.48**	**73.14**	**33.84**	**52.54**	**64.91**$_{6.4\%\downarrow}$
	2.20	**79.05**	**73.70**	**47.87**	**74.56**	**76.63**	**72.77**	**34.24**	**47.73**	**63.29**$_{8.0\%\downarrow}$
	2.05	**79.16**	**73.06**	**48.38**	**80.58**	**74.95**	**71.27**	**31.79**	**46.80**	**63.25**$_{8.0\%\downarrow}$
	1.94	**76.88**	**68.48**	**45.48**	**75.23**	**72.05**	**72.61**	**31.16**	**40.93**	**60.35**$_{10.9\%\downarrow}$
	1.81	**76.93**	**66.67**	**43.60**	**75.50**	**70.50**	**69.85**	**28.68**	**40.71**	**59.06**$_{12.2\%\downarrow}$
	1.69	**75.41**	**64.14**	**40.61**	**68.96**	**67.01**	**68.03**	**28.04**	**37.14**	**56.17**$_{15.1\%\downarrow}$
	1.57	**72.42**	**62.46**	**37.88**	**73.55**	**63.17**	**66.38**	**26.80**	**32.25**	**54.49**$_{16.8\%\downarrow}$

TABLE 3
Selected MoE-LLMs/VLMs and model configurations. Param.: the total parameter size (we set 16-bit as one parameter)(in LLM), Act Param.: activated parameter size per-token (in LLM); B: decoder block number, H: hidden dimension, E: expert number.

	Model	Param.	Act Param.	B	H	E
LLMs	Mixtral 8×7b	49b	13b	32	4096	8
	Mixtral 8×22b	141b	39b	56	6144	8
VLMs	DeepSeek-VL2-L	27B	4.1B	30	2569	72
	DeepSeek-VL2-S	16B	2.4B	26	2048	64
	DeepSeek-VL2-T	3B	0.6B	11	1792	64

effective strategy [29], [33]. We also adopt this as a comparative baseline for expert bit-width allocation. Fig. 9 and Fig. 10 include three benchmark curves: Hessian, Frobenius norm (F-norm), and PMQ. For experts bit-width allocation, both F-norm and PMQ outperform Hessian, consistently delivering better performance across various bit-widths. When the average bit-width exceeds 2-bit, the PPL curves of F-norm and PMQ appear similar; however, below 2-bit, PMQ's advantage becomes increasingly significant. This suggests that PMQ is particularly effective in identifying optimal mixed-precision strategies for MoE under ultra-low bit-width settings.

4.2.2 *Comparison of Mixed-Precision Quantization*

We conducted a comprehensive evaluation of the performance of *PMQ* in the ultra-low bit-width range. GPTQ was used as the baseline for uniform bit-width quantization, denoted as "Uni" in Tab. 2 and Tab. 4. Additionally, we

TABLE 4
Performance of quantized DeepSeek-VL2-L/S/T on six general multimodal benchmarks. We deploy GPTQ as our baseline PTQ method for uniform quantization. "Uni" denotes the uniform quantization of 2-bit with GPTQ. ↓ gives the accuracy loss between the quantized results and the original 16-bit model. The average results are calculated from MMBench, MMStar, MMMU, AI2D, and OCRBench.

Model	Method	Bits	MMBench	MMStar	MME	MMMU	AI2D	OCRBench	Avg.% ↑
DeepSeek-VL2-L		16.00	82.99	62.07	1644.93	54.67	82.32	81.30	72.67
	Uni	3.00	82.64	60.47	1637.85	48.67	82.90	90.80	71.01$_{1.6\%\downarrow}$
	Uni	2.00	60.91	49.00	1480.39	40.22	68.36	74.50	58.60$_{14.1\%\downarrow}$
	Hessian [29]	2.57	80.71	60.00	1617.23	45.38	78.25	74.61	67.79$_{4.7\%\downarrow}$
		2.08	72.01	56.93	1578.29	42.00	75.37	69.11	61.58$_{11.1\%\downarrow}$
		1.59	65.89	50.12	1580.33	40.00	73.30	63.57	58.58$_{14.1\%\downarrow}$
	PMQ	2.57	**82.56**	**60.00**	**1628.42**	**47.33**	**82.29**	**80.80**	**70.60**$_{2.1\%\downarrow}$
		2.08	**79.21**	**56.93**	**1611.77**	**44.00**	**81.31**	**78.30**	**67.95**$_{4.7\%\downarrow}$
		1.59	**74.83**	**54.30**	**1512.71**	**46.67**	**78.17**	**73.70**	**65.35**$_{7.3\%\downarrow}$
DeepSeek-VL2-S		16.00	78.61	56.73	1654.30	47.33	79.05	83.50	69.04
	Uni	3.00	78.69	54.00	1669.77	42.00	77.42	81.80	66.78$_{2.3\%\downarrow}$
	Uni	2.00	34.62	33.47	1247.61	23.33	48.77	57.00	39.44$_{29.6\%\downarrow}$
	Hessian [29]	2.58	72.88	52.27	1613.41	40.67	67.84	74.31	61.79$_{7.3\%\downarrow}$
		2.07	66.67	48.83	1498.27	36.77	67.53	72.52	58.46$_{10.6\%\downarrow}$
		1.58	57.28	41.05	1378.27	31.62	67.77	69.24	53.59$_{15.5\%\downarrow}$
	PMQ	2.58	**74.05**	**54.33**	**1674.45**	**43.67**	**69.53**	**76.70**	**63.66**$_{5.4\%\downarrow}$
		2.07	**69.16**	**51.27**	**1576.86**	**42.33**	**72.53**	**76.70**	**62.40**$_{6.6\%\downarrow}$
		1.58	**63.32**	**48.47**	**1470.23**	**35.77**	**67.26**	**71.30**	**57.23**$_{11.8\%\downarrow}$
DeepSeek-VL2-T		16.00	72.08	49.47	1554.01	39.89	74.77	80.30	63.30
	Uni	3.00	67.10	46.53	1560.91	38.33	70.85	77.4	60.04$_{3.3\%\downarrow}$
	Uni	2.00	1.12	1.00	448.38	0.00	4.93	0.00	1.41$_{61.9\%\downarrow}$
	Hessian [29]	2.59	66.68	42.36	1566.27	29.79	69.20	70.02	55.61$_{7.7\%\downarrow}$
		2.12	54.33	39.06	1490.27	30.00	57.46	71.03	50.58$_{12.7\%\downarrow}$
		1.63	25.51	30.02	998.70	15.03	30.00	66.56	33.42$_{29.88\%\downarrow}$
	PMQ	2.59	**70.10**	**46.93**	**1564.56**	**34.00**	**72.15**	**74.10**	**59.45**$_{3.8\%\downarrow}$
		2.12	**61.42**	**39.73**	**1515.26**	**31.33**	**63.99**	**70.90**	**53.47**$_{9.8\%\downarrow}$
		1.63	**25.51**	**34.67**	**1230.16**	**25.52**	**39.44**	**63.10**	**37.65**$_{25.7\%\downarrow}$

compared PMQ against a recent mixed-precision approach for MoE-based large language models (MoE-LLMs), known as the Block Sparsity Predictor (BSP) [6]. The experiments were performed on Mixtral 8×7b and the DeepSeek-VL2 models (L/S/T variants), with the average selectable expert bit-widths set between 1.5 and 2.5 bits. The final results, presented as the average bit-widths across all language backbones, are reported in the tables. As shown in Tab. 2 and Tab. 4, for both MoE-LLMs and MoE-VLMs, uniform precision quantization results in average performance losses of 1.3–3.3% for 3-bit models. However, for 2-bit models, the performance degradation becomes significantly more pronounced, with losses approaching 30%. This highlights the substantial challenges of maintaining model accuracy using existing uniform precision quantization methods under ultra-low bit-width settings.

Under a 2.54-bit setting, the average accuracy of the BSP algorithm is only 49.07%. In contrast, the proposed *PMQ* algorithm achieves an accuracy of 67.50%, exceeding BSP by 18.4% and only falling off the 16-bit Mixtral 8×7b by 3.8%. Notably, *PMQ* can maintain an accuracy of 54.49% at 1.57-bit, even outperforming BSP at 2.54-bit by 5.4%. Methods based on the Hessian algorithm, as shown in Tab. 2, consistently underperform compared to *PMQ* across different bit-widths. Specifically, while the Hessian algorithm lags by a marginal 0.2% at 2.54 bits, *PMQ* demonstrates a more pronounced advantage below 2 bits, leading by 9.4% at 1.57 bits. Further insights are provided in Tab. 4, where the *PMQ* algorithm achieves the best performance under identical low-bit settings across DeepSeek-VL2 models with varying parameter scales. For example, in the DeepSeek-VL2-L model, *PMQ* achieves a score of 70.60% at 2.57 bits, only 2.1% lower than the fp16 baseline, while reducing the model size to just 16% of the original. Additionally, it outperforms the Hessian-based algorithms. Notably, as the model size increases, quantization loss decreases, a trend corroborated by the curves in Fig. 2, which collectively indicate that compressed large-parameter MoE-VLMs provide greater benefits compared to small full-precision models.

4.2.3 *Pareto Advantage of PMQ*

When compressing models to an ultra-low bit-width range of 1.5–2.5 bits, we can flexibly balance the required bit-width, model size, and performance demands. Therefore, exploring the compression Pareto frontier under different bit-widths becomes particularly critical [53]. Fig. 11 and Fig. 12 compare the proposed PMQ strategy with other random mixed-precision strategies across various bit-widths in the low-bit

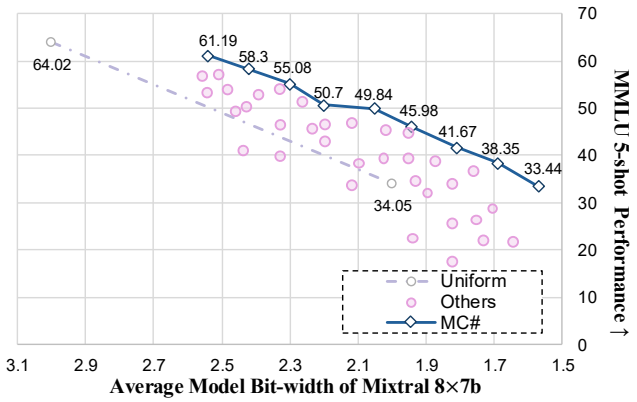

Fig. 11. Pareto curve of performance-precision trade-offs on Mixtral 8×7b model. "Others" denotes the other mixed-precision method, such as random configuration and other method in Fig. 9.

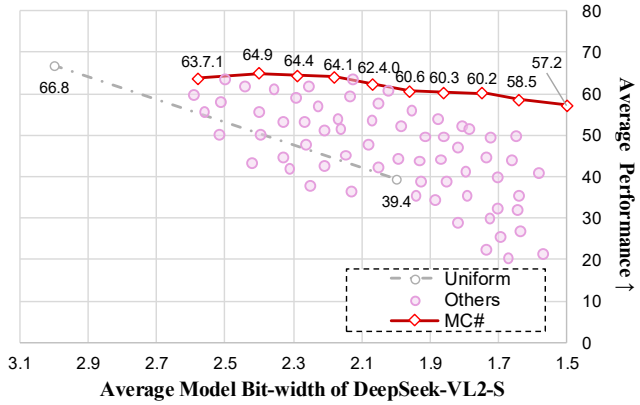

Fig. 12. Pareto curve of performance-precision trade-offs on DeepSeek-VL2-S model. "Others" denotes the other mixed-precision method, such as random configuration and other method in Fig. 10.

TABLE 5
Ablation evaluation of *PMQ* and *OTP* for MMoE-LLMs and MoE-VLMs. "Params" denotes the parameter size, and "Act Params" is the averaged activated parameters for one token. The parameter calculation of the compressed model includes the compressed weights and quantizer parameters (e.g., scaling factor and zero factor for dequantization). We carry out the average activated parameter size and speedup on the C4 and M4 datasets. 16-bit Mixtral 8×7b uses 2 A100-80GB GPUs, and Mixtral 8×22b uses 4. Other quantized MoE models are tested on one A100-80GB GPU.

LLMs	Bits	PMQ	OTP	Uni	LM-Eval% ↑	VLM-Eval% ↑	Params.(GB)	Act Params.(GB)	Speedup
Mixtral 8×7b	16.00	-	-	-	71.29	-	96.80	26.31	1.00×
	2.00	-	-	✓	42.67	-	13.61	3.70	1.72×
	2.05	✓	-	-	63.25	-	13.41	3.73	1.67×
	2.05	✓	✓	-	**62.68**	-	**13.41**	**3.23**	**1.80×**
Mixtral 8×22b	16.00	-	-	-	76.33	-	281.24	76.49	1.00×
	2.00	-	-	✓	50.44	-	38.08	10.35	1.95×
	2.05	✓	-	-	67.94	-	38.35	10.42	1.80×
	2.05	✓	✓	-	**66.50**	-	**38.35**	**9.03**	**1.87×**
DeepSeek-VL2-S	16.00	-	-	-	-	69.04	32.39	4.95	1.00×
	2.00	-	-	✓	-	39.43	4.54	0.81	1.62×
	2.07	✓	-	-	-	62.40	4.59	0.84	1.62×
	2.07	✓	✓	-	-	**60.92**	**4.59**	**0.52**	**1.92×**
DeepSeek-VL2-L	16.00	-	-	-	-	72.67	54.95	8.27	1.00×
	2.00	-	-	✓	-	58.60	7.85	1.18	1.78×
	2.08	✓	-	-	-	67.95	7.92	1.19	1.77×
	2.08	✓	✓	-	-	**66.42**	**7.92**	**0.79**	**1.86×**

range. The results demonstrate that, for both MoE-LLMs and MoE-VLMs, PMQ consistently achieves the optimal Pareto frontier under different bit-widths. Furthermore, we observed that due to the more pronounced sparsity of MoE-VLMs (as observed in Fig. 5), the Pareto curve for DeepSeek-VL2-S is notably flatter. This indicates that MoE-VLMs are inherently more suitable for mixed-precision compression.

4.3 Experiment on Online Top-any Pruning

In the pre-loading phase, *PMQ* enables the compression of MoE-LLMs to an exceptionally low bit-width range. Furthermore, during the inference phase, we apply the *OTP* outlined in Sec. 3.4 to the quantized MoE model, further enhancing the efficiency of real-time inference for lightweight models.

TABLE 6
Ablation of the combination of **MC#**. "Pruning Ratio" denotes the average expert's pruning percentage during the evaluation. We evaluate the MoE-LLMs on the WikiText2 dataset with PPL↓, MoE-VLMs on MMBench, MMStar, MMMU, AI2D, OCRBench with average score ↑.

Model	Method	Pruning Ratio(%)	Bits	PPL↓	Score↑
Mixtral 8×7b	PMQ	0	2.05	5.91	-
	PMQ	0	1.69	7.78	-
	PMQ+ODP [1]	21.27	2.05	6.62	-
	PMQ+OTP	**23.10**	**2.05**	**6.45**	-
DeepSeek-VL2-S	PMQ	0	2.07	-	62.40
	PMQ	0	1.64	-	58.46
	PMQ+random	16.67	2.07	-	52.69
	PMQ+OTP	**33.51**	**2.07**	-	**60.92**

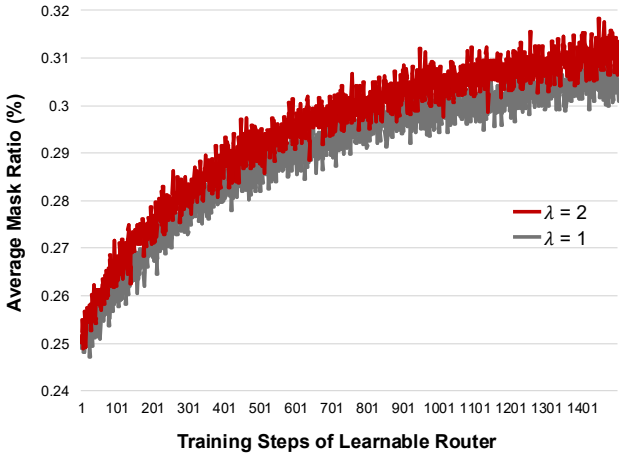

Fig. 13. Ablation of mask ratio during training. We train our proposed learnable router with the DeepSeek-VLs2-S model under different λ.

4.3.1 Sparsity Constrain

We first observed the impact of different mask sparsity parameters, denoted as λ, on the training process. As shown in Fig. 13, due to the gradient direction constraints introduced in Eq. 14, the model tends to adopt a higher mask ratio while minimizing the distillation loss as much as possible. When $\lambda = 1$, the model progressively increases the average mask ratio during training to approximately 30%. As λ increases (from 1 to 2 in Fig. 13), the mask ratio also rises accordingly, which aligns with the definition in Eq. 14. All OTP results reported in this study are based on $\lambda = 1$.

4.3.2 Comparison of Online Experts Pruning Methods

As shown in Tab. 6, the OTP method achieves a PPL of 6.45 on a 2.05-bit model with an additional expert masking ratio of 23.10%, whereas the rule-based ODP [1] only reaches a PPL of 6.62 with a 21.27% masking ratio. Moreover, as the k value in top-k increases within MoE-VLMs, rule-based methods like ODP struggle to perform effective pruning across the numerous possible expert combinations. In contrast, OTP enables dynamic expert masking through a flexible, soft, and learnable approach. For instance, OTP achieves dynamic pruning of 33.51% of experts in the 2.07-bit DeepSeek-VL2-S model, with only a 1.5% drop in the multimodal benchmark score. Interestingly, we observed that rather than further compressing model parameter size (from 2.07-bit to 1.64-bit), the PMQ+OTP configuration activates fewer total parameters during dynamic inference, resulting in less performance drop.

4.3.3 Performance on Challenging Language Benchmarks

In this section, we expand our experiments on more challenging datasets in Tab. 7, considering the importance of performance testing on more complex long text or reasoning capabilities of compressed MoE model [54]–[56]. We have observed that in challenging tasks like GSM8K, HumanEval, and long-context Needle-in-a-haystack, the performance drop of model compression becomes more pronounced. This phenomenon holds true in other MoE-LLM compression methods [8], [11], [14], [33] as well. However, our *PMQ* method, compared to the latest method like BSP [6] and HAWQ [29] with Hessian-based approaches for MoE-LLM, is still able to maintain state-of-the-art performance.

TABLE 7
Comparison of different mixed-precision quantization methods on challenging benchmarks. NIAH denotes the task in Needle-in-a-haystack, which is a more challenging task for evaluating long-context ability.

Method	Bits	GSM8K↑	HumanEval (p@10) ↑	NIAH↑
	16.00	58.30	59.15	100.00
Uniform	3.00	38.13	29.88	98.48
Uniform	2.00	0.00	0.00	0.00
BSP	2.54	4.25	3.21	42.21
Hessian	2.54	33.59	25.49	100.00
Hessian	2.05	17.24	7.84	93.45
PMQ	2.54	**37.67**	**29.34**	**100.00**
PMQ+OTP	2.54	**36.44**	**27.92**	**100.00**
PMQ	2.05	**19.97**	**11.83**	**100.00**
PMQ+OTP	2.05	**20.91**	**11.71**	**100.00**

TABLE 8
Latency comparison of MoE and dense LLM under different hardware platforms.

Model	GPU	Loading Memory	Token/s
Mixtral 8×7b	2×A100	96.8 GB	23
Mixtral 8×7b	1×3090	OOM	-
MC# 2.54-bit	1×3090	16.4 GB	53
DeepSeek-VL2-L	1×A100	55.0 GB	36
DeepSeek-VL2-L	1×3090	OOM	-
MC# 2.59-bit	1×3090	8.9 GB	65

4.4 Memory Saving and Inference Efficiency

Tab. 5 provides a detailed overview of the proposed **MC#** framework, including memory compression, speed benchmarks, and average performance results (through LM-Eval and VLMEval). We utilized the HQQ [50] tool for quantized weight storage and dequantization, and specifically designed a 1-bit transformation format for storing binary weights. For MoE-LLMs, we sampled 128 data points with a token length of 2048 from the C4 dataset. For MoE-VLMs, we randomly sampled 128 VQA data from the M4 dataset. The speedup was ultimately calculated based on the average token generation time. Following the application of PMQ's 2.05-bit quantization, the memory usage of the Mixtral 8×7b model was reduced from 96.8GB to 13.41GB, while DeepSeek-VL2-S was compressed from 32.39GB to 4.59GB. During dynamic inference, OTP further reduced activated parameters by approximately 20–33%, with an average accuracy loss of only about 1%. Additionally, PMQ, leveraging the HQQ-based quantization framework and ATEN kernels, achieved a 1.6–2x speedup, while OTP provided an additional 10–20% improvement in overall inference speed. Similar trends were observed across MoE-LLMs and MoE-VLMs of varying scales, further highlighting the effectiveness and scalability of the proposed approach.

4.4.1 Inference Latency

In Tab. 8, we report the empirical deployment speedups achieved by our MC method across diverse hardware platforms. The gains in **MC#** stem from static compression during the *PMQ* phase and from CUDA-kernel adaptations (based on HQQ) together with *OTP*. All throughput measurements were obtained with an input sequence length of 2048 tokens and an output length of 512 tokens. On an RTX 3090 GPU, MC-MoE with aggressive compression attains an average generation rate of 52 tokens/s, offering a cost-efficient deployment. In this setting, the compressed MoE LLM surpasses a dense LLM in memory footprint, accuracy, and speed.

5 Conclusion

MoE represents a promising framework of sparse models for multimodal understanding through scaling up the model capacity. However, the memory demands and redundancy among experts pose significant challenges for their practical implementation. In this work, we propose **MC#**, a mixture compression strategy based on the imbalance of significance among experts. This method co-designs the *Pre-Loading Mixed-Precision Quantization (PMQ)* and *Online Top-any Pruning (OTP)* approach, allowing MoE models to be compressed to an ultra-low bit-width while maintaining exceptional memory and parameter efficiency, as well as knowledgeable performance. And our mixed-precision strategy is orthogonal to various quantization techniques. Comprehensive experiments validate the effectiveness of our mixture compression, revealing that highly compressed MoE-LLMs and MoE-VLMs can even outperform equal-size full-precision dense LLMs, thereby improving the feasibility of MoE compression. Future work will focus on adapting this strategy for multimodal applications and optimizing it for specific hardware platforms.

Acknowledgements. This work has been supported in part by Hong Kong Research Grant Council - Early Career Scheme (Grant No. 27209621), General Research Fund Scheme (Grant No. 17202422, 17212923), Theme-based Research (Grant No. T45-701/22-R), the Innovation and Technology Fund (Mainland-Hong Kong Joint Funding Scheme, MHP/053/21), and the Shenzhen-Hong Kong-Macau Technology Research Program (SGDX20210823103537034). This research is also supported in part by National Key R&D Program of China (2022ZD0115502), National Natural Science Foundation of China (NO. 62461160308, U23B2010), "Pioneer" and "Leading Goose" R&D Program of Zhejiang (No. 2024C01161).

References

[1] W. Huang, Y. Liao, J. Liu, R. He, H. Tan, S. Zhang, H. Li, S. Liu, and X. Qi, "Mc-moe: Mixture compressor for mixture-of-experts llms gains more," *arXiv preprint arXiv:2410.06270*, 2024.

[2] N. Muennighoff, L. Soldaini, D. Groeneveld, K. Lo, J. Morrison, S. Min, W. Shi, P. Walsh, O. Tafjord, N. Lambert *et al.*, "Olmoe: Open mixture-of-experts language models," *arXiv preprint arXiv:2409.02060*, 2024.

[3] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand *et al.*, "Mixtral of experts," *arXiv preprint arXiv:2401.04088*, 2024.

[4] Z. Wu, X. Chen, Z. Pan, X. Liu, W. Liu, D. Dai, H. Gao, Y. Ma, C. Wu, B. Wang *et al.*, "Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding," *arXiv preprint arXiv:2412.10302*, 2024.

[5] B. Lin, Z. Tang, Y. Ye, J. Cui, B. Zhu, P. Jin, J. Huang, J. Zhang, Y. Pang, M. Ning *et al.*, "Moe-llava: Mixture of experts for large vision-language models," *arXiv preprint arXiv:2401.15947*, 2024.

[6] P. Li, X. Jin, Y. Cheng, and T. Chen, "Examining post-training quantization for mixture-of-experts: A benchmark," *arXiv preprint arXiv:2406.08155*, 2024.

[7] Z. Chi, L. Dong, S. Huang, D. Dai, S. Ma, B. Patra, S. Singhal, P. Bajaj, X. Song, X.-L. Mao *et al.*, "On the representation collapse of sparse mixture of experts," *Advances in Neural Information Processing Systems*, vol. 35, pp. 34 600–34 613, 2022.

[8] X. Lu, Q. Liu, Y. Xu, A. Zhou, S. Huang, B. Zhang, J. Yan, and H. Li, "Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models," *arXiv preprint arXiv:2402.14800*, 2024.

[9] Y. Koishekenov, A. Berard, and V. Nikoulina, "Memory-efficient nllb-200: Language-specific expert pruning of a massively multilingual machine translation model," *arXiv preprint arXiv:2212.09811*, 2022.

[10] Y. J. Kim, A. A. Awan, A. Muzio, A. F. C. Salinas, L. Lu, A. Hendy, S. Rajbhandari, Y. He, and H. H. Awadalla, "Scalable and efficient moe training for multitask multilingual models," *arXiv preprint arXiv:2109.10465*, 2021.

[11] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, "Gptq: Accurate post-training quantization for generative pre-trained transformers," *arXiv preprint arXiv:2210.17323*, 2022.

[12] A. Tseng, J. Chee, Q. Sun, V. Kuleshov, and C. De Sa, "Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks," *arXiv preprint arXiv:2402.04396*, 2024.

[13] M. Chen, W. Shao, P. Xu, J. Wang, P. Gao, K. Zhang, Y. Qiao, and P. Luo, "Efficientqat: Efficient quantization-aware training for large language models," *arXiv preprint arXiv:2407.11062*, 2024.

[14] W. Shao, M. Chen, Z. Zhang, P. Xu, L. Zhao, Z. Li, K. Zhang, P. Gao, Y. Qiao, and P. Luo, "Omniquant: Omnidirectionally calibrated quantization for large language models," *arXiv preprint arXiv:2308.13137*, 2023.

[15] V. Egiazarian, A. Panferov, D. Kuznedelev, E. Frantar, A. Babenko, and D. Alistarh, "Extreme compression of large language models via additive quantization," *arXiv preprint arXiv:2401.06118*, 2024.

[16] B. Liao and C. Monz, "Apiq: Finetuning of 2-bit quantized large language model," *arXiv preprint arXiv:2402.05147*, 2024.

[17] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[18] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang *et al.*, "A survey on evaluation of large language models," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 3, pp. 1–45, 2024.

[19] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[20] H. Lu, W. Liu, B. Zhang, B. Wang, K. Dong, B. Liu, J. Sun, T. Ren, Z. Li, H. Yang *et al.*, "Deepseek-vl: towards real-world vision-language understanding," *arXiv preprint arXiv:2403.05525*, 2024.

[21] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.

[22] L. Yun, Y. Zhuang, Y. Fu, E. P. Xing, and H. Zhang, "Toward inference-optimal mixture-of-expert large language models," *arXiv preprint arXiv:2404.02852*, 2024.

[23] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[24] Z. Zhou, X. Ning, K. Hong, T. Fu, J. Xu, S. Li, Y. Lou, L. Wang, Z. Yuan, X. Li *et al.*, "A survey on efficient inference for large language models," *arXiv preprint arXiv:2404.14294*, 2024.

[25] X. Zhu, J. Li, Y. Liu, C. Ma, and W. Wang, "A survey on model compression for large language models," *arXiv preprint arXiv:2308.07633*, 2023.

[26] J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, G. Xiao, X. Dang, C. Gan, and S. Han, "Awq: Activation-aware weight quantization for on-device llm compression and acceleration,"

Proceedings of Machine Learning and Systems, vol. 6, pp. 87–100, 2024.
[27] S. Li, Y. Hu, X. Ning, X. Liu, K. Hong, X. Jia, X. Li, Y. Yan, P. Ran, G. Dai *et al.*, "Mbq: Modality-balanced quantization for large vision-language models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 4167–4177.
[28] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, "Smoothquant: Accurate and efficient post-training quantization for large language models," in *International Conference on Machine Learning*. PMLR, 2023, pp. 38 087–38 099.
[29] Z. Dong, Z. Yao, D. Arfeen, A. Gholami, M. W. Mahoney, and K. Keutzer, "Hawq-v2: Hessian aware trace-weighted quantization of neural networks," *NeurIPS*, vol. 33, pp. 18 518–18 529, 2020.
[30] Y. Shang, Z. Yuan, Q. Wu, and Z. Dong, "Pb-llm: Partially binarized large language models," *arXiv preprint arXiv:2310.00034*, 2023.
[31] W. Huang, Y. Liu, H. Qin, Y. Li, S. Zhang, X. Liu, M. Magno, and X. Qi, "Billm: Pushing the limit of post-training quantization for llms," *arXiv preprint arXiv:2402.04291*, 2024.
[32] T. Dettmers, R. Svirschevski, V. Egiazarian, D. Kuznedelev, E. Frantar, S. Ashkboos, A. Borzunov, T. Hoefler, and D. Alistarh, "Spqr: A sparse-quantized representation for near-lossless llm weight compression," *arXiv preprint arXiv:2306.03078*, 2023.
[33] W. Huang, H. Qin, Y. Liu, Y. Li, X. Liu, L. Benini, M. Magno, and X. Qi, "Slim-llm: Salience-driven mixed-precision quantization for large language models," *arXiv preprint arXiv:2405.14917*, 2024.
[34] Z. Liu, B. Oguz, C. Zhao, E. Chang, P. Stock, Y. Mehdad, Y. Shi, R. Krishnamoorthi, and V. Chandra, "Llm-qat: Data-free quantization aware training for large language models," *arXiv preprint arXiv:2305.17888*, 2023.
[35] H. Guo, P. Greengard, E. P. Xing, and Y. Kim, "Lq-lora: Low-rank plus quantized matrix decomposition for efficient language model finetuning," *arXiv preprint arXiv:2311.12023*, 2023.
[36] W. Huang, X. Ma, H. Qin, X. Zheng, C. Lv, H. Chen, J. Luo, X. Qi, X. Liu, and M. Magno, "How good are low-bit quantized llama3 models? an empirical study," *arXiv preprint arXiv:2404.14047*, 2024.
[37] W. Kwon, S. Kim, M. W. Mahoney, J. Hassoun, K. Keutzer, and A. Gholami, "A fast post-training pruning framework for transformers," *NeurIPS*, vol. 35, pp. 24 101–24 116, 2022.
[38] I. Hubara, B. Chmiel, M. Island, R. Banner, J. Naor, and D. Soudry, "Accelerated sparse neural training: A provable and efficient method to find n: m transposable masks," *NeurIPS*, vol. 34, pp. 21 099–21 111, 2021.
[39] E. Frantar and D. Alistarh, "Sparsegpt: Massive language models can be accurately pruned in one-shot," in *International Conference on Machine Learning*. PMLR, 2023, pp. 10 323–10 337.
[40] M. Sun, Z. Liu, A. Bair, and J. Z. Kolter, "A simple and effective pruning approach for large language models," *arXiv preprint arXiv:2306.11695*, 2023.
[41] G. Fang, H. Yin, S. Muralidharan, G. Heinrich, J. Pool, J. Kautz, P. Molchanov, and X. Wang, "Maskllm: Learnable semi-structured sparsity for large language models," *arXiv preprint arXiv:2409.17481*, 2024.
[42] E. Liu, J. Zhu, Z. Lin, X. Ning, M. B. Blaschko, S. Yan, G. Dai, H. Yang, and Y. Wang, "Efficient expert pruning for sparse mixture-of-experts language models: Enhancing performance and reducing inference costs," *arXiv preprint arXiv:2407.00945*, 2024.
[43] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.
[44] F. Li, R. Zhang, H. Zhang, Y. Zhang, B. Li, W. Li, Z. Ma, and C. Li, "Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models," *arXiv preprint arXiv:2407.07895*, 2024.
[45] T. Gale, D. Narayanan, C. Young, and M. Zaharia, "Megablocks: Efficient sparse training with mixture-of-experts," *Proceedings of Machine Learning and Systems*, vol. 5, pp. 288–304, 2023.
[46] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European conference on computer vision*. Springer, 2016, pp. 525–542.
[47] H. Huang, N. Ardalani, A. Sun, L. Ke, H.-H. S. Lee, A. Sridhar, S. Bhosale, C.-J. Wu, and B. Lee, "Towards moe deployment: Mitigating inefficiencies in mixture-of-expert (moe) inference," *arXiv preprint arXiv:2303.06182*, 2023.
[48] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1389–1397.
[49] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt, "Measuring mathematical problem solving with the math dataset," *arXiv preprint arXiv:2103.03874*, 2021.
[50] H. Badri and A. Shaji, "Towards 1-bit machine learning models," March 2024. [Online]. Available: https://mobiusml.github.io/1bit_blog/
[51] L. Gao, J. Tow, B. Abbasi, S. Biderman, S. Black, A. DiPofi, C. Foster, L. Golding, J. Hsu, A. Le Noac'h *et al.*, "A framework for few-shot language model evaluation," *URL https://zenodo.org/records/10256836*, vol. 7, 2013.
[52] H. Duan, J. Yang, Y. Qiao, X. Fang, L. Chen, Y. Liu, X. Dong, Y. Zang, P. Zhang, J. Wang *et al.*, "Vlmevalkit: An open-source toolkit for evaluating large multi-modality models," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 11 198–11 201.
[53] Z. Liu, C. Zhao, H. Huang, S. Chen, J. Zhang, J. Zhao, S. Roy, L. Jin, Y. Xiong, Y. Shi *et al.*, "Paretoq: Scaling laws in extremely low-bit llm quantization," *arXiv preprint arXiv:2502.02631*, 2025.
[54] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano *et al.*, "Training verifiers to solve math word problems," *arXiv preprint arXiv:2110.14168*, 2021.
[55] Y. Bai, X. Lv, J. Zhang, H. Lyu, J. Tang, Z. Huang, Z. Du, X. Liu, A. Zeng, L. Hou *et al.*, "Longbench: A bilingual, multitask benchmark for long context understanding," *arXiv preprint arXiv:2308.14508*, 2023.
[56] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. D. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman *et al.*, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.

Wei Huang (Graduate Student Member, IEEE) is currently a Ph.D. at the Department of Electrical and Electronic Engineering (EEE) of the University of Hong Kong, as a member of the Computer Vision and Machine Intelligence (CVMI) Lab and Wearable, Intelligent and Soft Electronics (WISE) Lab. He obtained his B.S. in computer science from the School of Computer Science and Engineering, Beihang University. His research interests include efficient multimodal LLMs, multimodality reasoning and wearable AI. He has published several papers on top journals and conferences, including ICML, ICLR, CVPR, etc.

Yue Liao is currently a research fellow at School of Computing, National University of Singapore. He received his PhD degree at School of Computer Science and Engineering, Beihang University. His research interests include multimodal understanding and embodied AI. He has published more than 30 papers at top journals and conferences, including T-PAMI, T-IP, NIPS, ICLR, CVPR, ICCV and ECCV, etc.

Yukang Chen is a research scientist in NVIDIA Research, working on efficient LLMs and VLMs. He received his PhD from the Chinese University of Hong Kong this year, specializing in efficient deep learning, LLMs, and computer vision. He has published over 30 papers in top conferences and journals, with 10 as the first author. His work has been featured in multiple oral presentations at top conferences such as ICLR and CVPR, and has accumulated over 5,000 citations on Google Scholar. His first-authored open-source projects have received over 6,000 stars on GitHub. Yukang was selected as a finalist candidate for the ByteDance Fellowship and has achieved winner/1st positions in renowned competitions and leaderboards, including Microsoft COCO, ScanNet, and nuScenes.

Jianhui Liu is currently a final year Ph.D student at the University of Hong Kong. He received his bachelor's degree from the School of Computer Science, Xidian University. His research interests include 3D scene understanding, 6D pose estimation and efficient model design. He has published more than 10 papers at top journals and conferences, including TKDR, NIPS, CVPR, ICCV, ICLR etc.

Haoru Tan is currently a second-year Ph.D student at the University of Hong Kong. He received his master's degree from the Chinese Academy of Sciences, Institute of Automation. He has also had internship experiences at institutions such as Baidu Research, Alibaba DAMO Academy, and Tencent Research. His research interests include data-centric AI, machine learning, and optimizations. He has published more than 10 papers at top journals and conferences, including NeurIPS, ICLR, ICCV, CVPR, IJCV, and T-PAMI.

Si Liu (Senior Member, IEEE) is currently a full professor at Institute of Artificial Intelligence, Beihang University. She is the recipient of the National Science Fund for Excellent Young Scholars. Her research interests include cross-modal multimedia intelligent analysis and classical computer vision tasks. She has published more than 60 cutting-edge papers and been cited over 13000 times on Google Scholar. She has won the Best Paper Awards of ACM MM 2021 and 2013, the Best Video Award of IJCAI 2021, and the Best Demo Award of ACM MM 2012. She is currently the associate editor of IEEE TMM, IEEE TCSVT, and CVIU, and she has served as the area chair of ICCV, CVPR, ECCV, ACM MM, and other top conferences many times.

Shiming Zhang (Member, IEEE) is currently an Assistant Professor at the Department of Electrical and Electronic Engineering (EEE) of the University of Hong Kong, leading the wearable, intelligent, and soft electronics (WISE) research group. Before that, he spent 3 years at the University of California, Los Angeles (UCLA), as a postdoctoral scholar (group leader on bioelectronics), and obtained his Ph.D. from École Polytechnique, Université de Montréal, Canada, and B.S./M.S. from Jilin University, China. He was recognized as a "Rising Star" by Advanced Materials (2024) and an "Emerging Investigator" by JMCC (2022) for his contributions to the interdisciplinary fields of soft semiconductor devices, electrochemistry, and hydrogel bioelectronics.

Shuicheng Yan (Fellow, IEEE) is a Distinguished Professor (Practice) at the School of Computing, National University of Singapore, and formerly served as the Group Chief Scientist at Sea Group, alongside several other notable industry engagements. He is a Fellow of the Singapore Academy of Engineering, AAAI, ACM, IEEE, and IAPR. His research focuses on computer vision, machine learning, and multimedia analysis. To date, he has published over 800 papers in top-tier international journals and conferences, achieving an H-index exceeding 140. He has also been recognized as one of the World's Highly Cited Researchers for ten years. In addition, his team has secured ten first-place or honorable-mention accolades in two flagship competitions, Pascal VOC and ImageNet (ILSVRC). Additionally, his team has earned more than ten best paper and best student paper awards, including a remarkable grand slam at ACM Multimedia, a premier conference in multimedia research, with three Best Paper Awards, two Best Student Paper Awards, and one Best Demo Award.

Xiaojuan Qi (Senior Member, IEEE) is currently an assistant professor at the Department of Electrical and Electronic Engineering (EEE) of the University of Hong Kong, leading the Computer Vision and Machine Intelligence Lab (CVMI Lab). Before that, she spent 1 year at the University of Oxford, UK, as a postdoctoral scholar with Prof. Philip Torr, and obtained her Ph.D. from The Chinese University of Hong Kong, Hong Kong, and B.S. from Shanghai Jiao Tong University, China. Her research encompasses the broad areas of Computer Vision, Deep Learning, and Artificial Intelligence. She has published more than 100 cutting-edge papers and has been cited over 38000 times on Google Scholar, and has also been named among 'IEEE AI's 10 to Watch for 2024' and '35 Innovators Under 35 for China by MIT Technology. She has served as the area chair of NeurIPS, ICCV, CVPR, ECCV, and other top conferences many times.