CAMERA: Multi-Matrix Joint Compression for MoE Models via Micro-Expert Redundancy Analysis

Yuzhuang Xu[1], Xu Han[2], Yuanchi Zhang[3], Yixuan Wang[1], Yijun Liu[1], Shiyu Ji[1], Qingfu Zhu[1], Wanxiang Che[1,*]

[1]Research Center for Social Computing and Interactive Robotics, Harbin Institute of Technology, Harbin, China
[2]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[3]WeChat AI, Tencent Inc, Beijing, China
{xyz, car}@ir.hit.edu.cn

Abstract

Large Language Models (LLMs) with Mixture-of-Experts (MoE) architectures are distinguished by their strong performance scaling with increasing parameters across a wide range of tasks, yet they also suffer from substantial computational and storage overheads. Notably, the performance gains of MoE models do not scale proportionally with the growth in expert parameters. While prior works attempt to reduce parameters via expert-level pruning, merging, or decomposition, they still suffer from challenges in both performance and computational efficiency. In this paper, we address these challenges by introducing **micro-expert** as a finer-grained compression unit that spans across matrices. We first establish a more fundamental perspective, viewing MoE layers as mixtures of micro-experts, and present CAMERA, a lightweight and training-free framework for identifying micro-expert redundancy. Our analysis uncovers significant variance in micro-expert contributions during decoding. Based on this insight, we further propose CAMERA-P, a structured micro-expert pruning framework, and CAMERA-Q, a mixed-precision quantization idea designed for micro-experts. Extensive experiments on nine downstream tasks show that CAMERA-P consistently outperforms strong baselines under pruning ratios ranging from 20% to 60%. Furthermore, CAMERA-Q achieves superior results under aggressive 2-bit quantization, surpassing existing matrix- and channel-level ideas. Notably, our method enables complete micro-expert analysis of Qwen2-57B-A14B in less than 5 minutes on a single NVIDIA A100-40GB GPU.

Code — https://github.com/xuyuzhuang11/CAMERA
Extended version — https://arxiv.org/abs/2508.02322

1 Introduction

Large Language Models (LLMs) based on Mixture-of-Experts (MoE) architecture leverage sparse Feed-Forward Network (FFN) structures to enable efficient model scaling, where each time only a subset of experts is activated by a router (Shazeer et al. 2017; Fedus, Zoph, and Shazeer 2022). This design facilitates the scaling of LLMs to hundreds of billions of parameters. Many well-known open-source models, such as Qwen3-MoE (Yang et al. 2025), Deepseek-MoE (Liu et al. 2024a), Kimi-K2 (Team et al. 2025), and Mixtral-MoE (Jiang et al. 2024), adopt the MoE architecture and achieve impressive results on downstream tasks. However, such substantial expansion in parameters does not bring proportional improvements in model capability, while also coupling with prohibitive computational and storage overheads—highlighting the structural redundancy inherent in MoE designs.

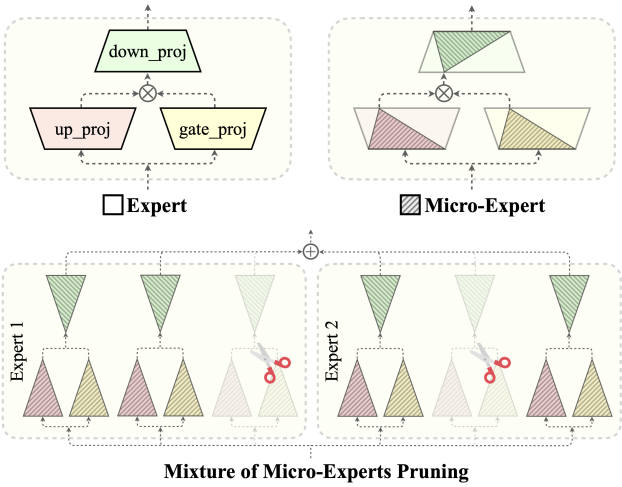

Figure 1: Transition from Experts to Micro-Experts. The lower part illustrates the structure of the mixture of micro-experts and the corresponding pruning strategy.

Pruning is a widely used strategy to reduce redundancy in MoE models. Existing methods for reducing expert parameters primarily fall into two categories: expert pruning and expert merging. Expert pruning removes either all or part of the parameters within each expert (Lu et al. 2024; Yang et al. 2024). Full pruning inevitably results in severe information loss, while partial pruning often yields suboptimal performance due to the lack of reliable measures for intra-expert importance. Expert merging, on the other hand, seeks to mitigate information loss by assuming functional similarity among experts (Yang et al. 2024; Li et al. 2024). However, this assumption rarely holds in practice, thereby limiting the effectiveness of merging-based strategies. An emerging direction exemplified by D^2-MoE attempts to relax this assumption by combining expert merging with delta com-

*Corresponding author

pression. Specifically, it constructs a shared base weight and expert-specific delta weights, both compressed using low-rank approximation (Gu et al. 2025; Ai et al. 2025).

Although recent studies suggest that partial expert pruning is more effective than full expert removal (Xie et al. 2024), especially in modern MoE models with a large number of experts, challenges remain due to limited understanding of expert behavior within MoE layers. In particular, it is difficult to identify essential parameters and determine optimal retention ratios for each expert. A further limitation of most existing methods is that they compress each matrix in the expert independently, overlooking the functional dependencies among different matrices. To shed light on the internal mechanism of MoE layers, we introduce a finer-grained structural perspective by viewing each MoE layer as a mixture of micro-experts, where each micro-expert is jointly defined by the three transformations—`up_proj`, `gate_proj` and `down_proj`, as shown in Figure 1. Our analysis reveals that the output of an MoE layer is a linear combination of all micro-experts, whose relative importance varies significantly. This insight forms the critical basis for the compression strategies proposed in this work.

We frame micro-expert pruning as the problem of removing a fixed proportion (e.g., 20%) of micro-experts while minimizing degradation in decoding performance. As we detail later, this is an NP-hard problem that cannot be solved exactly within practical time constraints. To address this challenge, we propose CAMERA, an efficient and accurate approximation algorithm for estimating the importance ranking of micro-experts in MoE layers. Building on this analysis, we propose CAMERA-P, a structured pruning framework that jointly prunes redundant micro-experts across the three FFN weight matrices, as shown in Figure 1. We further extend this idea to mixed-precision quantization by proposing CAMERA-Q, a novel micro-expert-aware partitioning scheme for assigning precision. Experiments show that CAMERA-P can prune MoE models (>50B) within 5 minutes on a single NVIDIA A100-40GB GPU—orders of magnitude faster than existing methods that often require hours of computation on multi-GPUs. Both CAMERA-P and CAMERA-Q consistently outperform strong baselines across nine zero-shot tasks and diverse MoE models, demonstrating superior efficiency and generalization.

Overall, we summarize our key contributions as follows:

- We propose CAMERA, a training-free and effective approximation algorithm that accurately ranks micro-experts by their importance in MoE layers, providing the foundation for micro-expert-based compression.
- We propose CAMERA-P, a structured pruning framework that jointly prunes across the matrices in each FFN, preserving functional integrity and coordination.
- We present CAMERA-Q, a novel micro-expert–aware mixed-precision quantization idea that allocates bit-widths based on micro-expert importance.
- Extensive experiments on mainstream MoE models and benchmarks demonstrate that both CAMERA-P and CAMERA-Q consistently outperform strong baselines, while achieving high scalability and efficiency.

2 Related Work

2.1 Model Compression

This work focuses on model weight compression. The primary approaches for compressing LLM weights include quantization, pruning, knowledge distillation, and low-rank approximation. Most quantization methods target post-training conversion of well-trained model weights into low-bit representations (Frantar et al. 2022; Xiao et al. 2023). The state-of-the-art quantization algorithms can reduce weights to 3-bit precision with negligible performance degradation (Liu et al. 2025; Shao et al. 2024). With the aid of carefully designed mixed-precision strategies, even sub-3-bit quantization can maintain performance comparable to the original model (Tseng et al. 2024; Xu et al. 2025). Pruning, on the other hand, reduces model size by eliminating less critical weights. Unstructured pruning offers maximum flexibility in weight removal and typically results in the least performance drop (Sun et al. 2024). Structured pruning imposes constraints on which weights can be pruned based on their location or structure, which may lead to slightly larger performance loss but enable actual inference speedup and make it more suitable for deployment (Frantar and Alistarh 2023). Knowledge distillation guided by loss minimization (Xu et al. 2024), and low-rank approximation typically via singular value decomposition (Ping et al. 2024), are also commonly integrated into broader compression frameworks. For further reading, please refer to survey (Zhou et al. 2024). As MoE becomes a dominant architecture for scaling LLMs, several compression techniques are developed specifically for MoE-based models.

2.2 Pruning Methods for MoE Models

Most pruning methods for MoE models focus on expert-level compression rather than on individual matrices. These fall mainly into two categories: direct pruning and expert merging (sometimes followed by pruning). Direct pruning methods like NAEE exhaustively evaluate expert combinations and retain the one yielding the lowest loss on calibration data (Lu et al. 2024). While effective on Mixtral-MoE, this brute-force approach does not scale well to modern sparse MoEs. MoE-I^2 (Yang et al. 2024) and MoE-Pruner (Xie et al. 2024) partially prune expert weights, but struggle to balance identifying important weights and achieving speedup. Expert merging methods fall into two subgroups. The first assumes functional similarity across experts, grouping and merging them based on activation frequency, output similarity, or knowledge distribution (e.g., MC-SMoE (Li et al. 2024), HC-SMoE (Chen et al. 2024), TAP (Zhang et al. 2024), EEP (Liu et al. 2024b)). This idealized assumption often limits performance. The second extracts common components from all experts (e.g., D^2-MoE (Gu et al. 2025), ResMoE (Ai et al. 2025), SubMoE (Li et al. 2025a), MoE-SVD (Li et al. 2025b)), then applies low-rank compression to approximate residuals. However, these approaches overlook the functional integrity across the three transformations in FFNs and often fail to preserve the most critical parameters. Our method, CAMERA-P, belongs to the partial pruning family but dif-

fers in two key ways. First, it performs fine-grained pruning at the micro-expert level, capturing cross-matrix coordination in FFNs. Second, it is powered by CAMERA, our fast and accurate importance estimator, addressing a major bottleneck of existing methods.

2.3 Quantization Methods for MoE Models

Most quantization work targets LLMs rather than MoEs, with MoE studies focusing mainly on assigning different bit-widths across experts. Studies like MC (Huang et al. 2025), MxMoE (Duanmu et al. 2025), and AFGQ (Xie et al. 2025) assign bit-widths based on a combination of activation frequency and weight sensitivity. In contrast, our CAMERA-Q is built on CAMERA, requiring no activation statistics, no pre-quantization, and no expensive evolutionary search.

3 Micro-Expert Redundancy Analysis

In this section, we first formalize the definition of micro-experts, followed by a general mathematical model that characterizes redundancy in MoE layers. We then highlight the challenges in analyzing such redundancy and derive the CAMERA algorithm as an efficient approximation, with provable and controllable error bounds.

3.1 From Expert to Micro-Expert

In the standard MoE architecture, each decoder layer comprises a self-attention layer and an MoE layer. The *i*-th expert in the MoE layer is typically defined as:

$$E_i\left(\mathbf{x}\right) = \mathbf{W}_i^{\text{down}}\left[\sigma\left(\mathbf{W}_i^{\text{gate}}\mathbf{x}\right)\cdot\mathbf{W}_i^{\text{up}}\mathbf{x}\right], \tag{1}$$

where $\mathbf{x}\in\mathbb{R}^{d_{\text{model}}}$ and $E_i(\mathbf{x})\in\mathbb{R}^{d_{\text{model}}}$ are the input/output hidden states, $\mathbf{W}^{\text{up/gate/down}}$ are the transformation matrices, and $\sigma(\cdot)$ is the SiLU activation. The MoE layer produces a weighted sum of N_E expert outputs, with weights $A_i(\mathbf{x})$ determined by the router (omit Top-*k*). This process is:

$$\mathbf{y} = \sum_i^{N_E} A_i\left(\mathbf{x}\right)\cdot E_i\left(\mathbf{x}\right). \tag{2}$$

We now adopt a microscopic perspective by decomposing each expert into micro-experts. Consider the weight matrices of a given expert. Let the *i*-th row of $\mathbf{W}^{\text{up}}$ and $\mathbf{W}^{\text{gate}}$ be denoted as $\mathbf{w}_i^{\text{up/gate}}$, and the *i*-th column of $\mathbf{W}^{\text{down}}$ as $\mathbf{w}_i^{\text{down}}$. We then define the *i*-th micro-expert as follows:

$$e_i\left(\mathbf{x}\right) = \mathbf{w}_i^{\text{down}}\left[\sigma\left(\mathbf{w}_i^{\text{gate}}\mathbf{x}\right)\cdot\mathbf{w}_i^{\text{up}}\mathbf{x}\right]. \tag{3}$$

The output of the MoE layer $\mathbf{y}=\sum_i^{N_e} A_i\left(\mathbf{x}\right)\cdot e_i\left(\mathbf{x}\right)$ can be viewed as a weighted combination of all N_e micro-experts, where the number of micro-experts $N_e = N_E\times d_{\text{ff}}$, and d_{ff} is the intermediate dimension of each expert. Note that $A_i\left(\mathbf{x}\right)$, $\sigma\left(\mathbf{w}_i^{\text{gate}}\mathbf{x}\right)$, and $\mathbf{w}_i^{\text{up}}\mathbf{x}$ are all scalars, so we denote

$$\phi_i = a_i\left(\mathbf{x}\right)\cdot\sigma\left(\mathbf{w}_i^{\text{gate}}\mathbf{x}\right)\cdot\mathbf{w}_i^{\text{up}}\mathbf{x}. \tag{4}$$

Based on this, Equation 2 can be rewritten as:

$$\mathbf{y} = \sum_i^{N_e}\phi_i\mathbf{w}_i^{\text{down}}. \tag{5}$$

For simplicity, we denote all $\mathbf{w}_i^{\text{down}}$ as $\mathbf{w}_i$ in the following paper. Clearly, the function of each micro-expert comprises two parts, represented by ϕ_i and $\mathbf{w}_i$ respectively. The $\{\mathbf{w}_i\}_{i=1}^{n_e}$ are fixed weight vectors, which we refer to as the **basis vector set**. This indicates that the output $\mathbf{y}$ of the MoE layer is actually a linear combination of the basis vectors. For different hidden states $\mathbf{x}$, the **combination coefficients** ϕ_i vary and are determined by $\mathbf{x}$.

3.2 Redundancy Problem

Let $(\mathbf{x},\mathbf{y})$ denote the input-output hidden state tuple of the MoE layer. Given any input $\mathbf{x}$, we aim to find a subset of micro-experts such that the decoding result $\hat{\mathbf{y}}$ based on this subset deviates minimally from the original $\mathbf{y}$. The micro-experts excluded in this process are considered to exhibit the highest redundancy. We introduce a calibration dataset $\{(\mathbf{x}_i,\mathbf{y}_i)\}_{i=1}^{n}$ as a proxy for the ideal hidden state space. For the *i*-th sample, we have:

$$\mathbf{y}_i = \phi_{i1}\mathbf{w}_1 + \phi_{i2}\mathbf{w}_2 + \cdots + \phi_{iN_e}\mathbf{w}_{N_e}. \tag{6}$$

This decoding process can be compactly expressed using matrix notation as $\mathbf{Y}=\mathbf{\Phi}\mathbf{W}$, where $\mathbf{\Phi}\in\mathbb{R}^{n\times N_e}$ is the activation coefficient matrix of all inputs $\mathbf{x}_i$ over all micro-experts, and $\mathbf{W}\in\mathbb{R}^{N_e\times d_{\text{model}}}$ is the matrix of basis vectors (i.e., the transpose of $\mathbf{W}^{\text{down}}$).

Now, to decode using a selected subset of m micro-experts, we aim to minimize the error between the resulting output and the original output $\mathbf{Y}$. Let S denote the index set of the selected micro-experts to be retained. This optimization problem can be formally written as:

$$\min_{S\subset[N_e],|S|=m}\|\mathbf{Y}-\mathbf{\Phi}_{:,S}\mathbf{W}_{S,:}\|_F^2. \tag{7}$$

This is similar to a class of problems known as Column Subset Selection Problems (CSSP). Such combinatorial optimization problems have been proven to be NP-hard, making it impossible to obtain exact solutions in polynomial time (Shitov 2021). Although general approximate solutions exist (Mahoney and Drineas 2009; Ordozgoiti, Canaval, and Mozo 2018), the large N_e still poses significant challenges for efficient hardware utilization. For instance, models like Mixtral-8×7B and Deepseek-MoE-16B have N_e on the order of 10^5. Even with approximation, the time and space costs for these models remain substantial.

3.3 CAMERA Algorithm

To approximately solve the optimization problem formulated in Equation 7, we first consider the impact of pruning a small subset of micro-experts on the output $\mathbf{Y}$. The following lemma provides an initial result.

Lemma Let S^C be the index set of removed micro-experts. Then the upper-bound of the decoding error ϵ on the calibration set is given by:

$$\epsilon_{\text{sup}} = \sum_{i\in S^C}\|\mathbf{\Phi}_{:,i}\|_2^2\|\mathbf{w}_i\|_2^2. \tag{8}$$

This lemma indicates that the decoding error upper-bound is related to both the combination coefficients and the basis

vectors. To minimize this upper-bound, we should prioritize pruning micro-experts with smaller norms of combination coefficients and basis vectors. The following definition provides a formal description of this intuition.

Definition The decoding-time energy $\mathcal{E}$ of the i-th micro-expert is proportional to the norm of its activation coefficient $\mathbf{\Phi}_{:,i}$ and basis vector $\mathbf{w}_i$, denoted as:

$$\mathcal{E}_i = \|\mathbf{\Phi}_{:,i}\|_2^2 \|\mathbf{w}_i\|_2^2. \tag{9}$$

Based on the definition of energy, we can rank all micro-experts and prioritize retaining the Top-$|S|$ highest-energy ones. Before presenting the specific algorithm, we first provide its tighter error bound.

Theorem Let $\hat{\mathbf{Y}}$ denote the decoding result using the Top-$|S|$ highest-energy micro-experts, and let $\mathbf{Y}^*$ denote the rank-$|S|$ SVD approximation of $\mathbf{Y}$. If $k = N_e - |S|$, the approximation error of $\hat{\mathbf{Y}}$ differs from the optimal SVD by only an $O(k)$-delta, i.e.,

$$\|\mathbf{Y} - \hat{\mathbf{Y}}\|_F^2 \leq \|\mathbf{Y} - \mathbf{Y}^*\|_F^2 + \delta\left(O(k)\right). \tag{10}$$

This theorem establishes a relationship with the optimal error bound. The above analysis shows that the energy-based estimation of micro-expert redundancy yields a controllable approximation error. All proofs in this section are given in the Appendix.

Based on this, we propose the CAMERA algorithm. Inspired by prior work, we aim to extend the definition of energy by additionally considering the effect of the maximum activation coefficient. The revised computation is given by:

$$\mathcal{E}_i = \left[(1-\alpha)\,\|\mathbf{\Phi}_{:,i}\|_2^2 + \alpha\|\mathbf{\Phi}_{:,i}\|_\infty^2\right] \cdot \|\mathbf{w}_i\|_2^2. \tag{11}$$

For a given MoE layer, we estimate the energy of all micro-experts on the calibration dataset using Equation 11 and rank them accordingly. This ranking reflects the importance of each micro-expert: the lower the energy, the more redundant the micro-expert is. Notably, CAMERA treats all micro-experts uniformly, without distinguishing which expert they belong to. Moreover, if shared experts exist, they do not need to be treated specially. The procedure for a single MoE layer is summarized in Algorithm 1.

Algorithm 1: CAMERA —Micro-Experts Ranking

Input: MoE layer with N_E experts, each with weights $\mathbf{W}_i^{\text{up}}$, $\mathbf{W}_i^{\text{gate}}$, $\mathbf{W}_i^{\text{down}}$; router $\mathbf{R}$; calibration dataset $(\mathbf{X}, \mathbf{Y})$; balance coefficient α
Output: Rank of all micro-experts

$\boldsymbol{\phi}^2, \boldsymbol{\phi}^\infty, \boldsymbol{\omega}^2 \leftarrow$ `Zeros`(shape=$N_E \cdot d_{\text{ff}}$)
$\mathbf{A} \leftarrow$ `Top&Norm`($\mathbf{RX}$)
for each expert $i = 1$ to N_E **do**
  $p, q \leftarrow i \cdot d_{\text{ff}},\ (i{+}1) \cdot d_{\text{ff}}$
  $\boldsymbol{\omega}^2_{p:q} \leftarrow$ `ColumnSum`($\mathbf{W}_i^{\text{down}} \cdot \mathbf{W}_i^{\text{down}}$)
  $\mathbf{\Lambda} \in \mathbb{R}^{n \times d_{\text{ff}}} \leftarrow \mathbf{A}_{:,i} \cdot \sigma\left(\mathbf{W}_i^{\text{gate}}\mathbf{X}\right) \cdot \mathbf{W}_i^{\text{up}}\mathbf{X}$
  $\boldsymbol{\phi}^2_{p:q} \leftarrow$ `ColumnSum`($\mathbf{\Lambda} \cdot \mathbf{\Lambda}$)
  $\boldsymbol{\phi}^\infty_{p:q} \leftarrow$ `ColumnMax`(`Abs`($\mathbf{\Lambda}$))^2
end for
$\mathcal{E} \leftarrow [(1-\alpha)\boldsymbol{\phi}^2 + \alpha\boldsymbol{\phi}^\infty] \cdot \boldsymbol{\omega}^2$
return `Argsort`($\mathcal{E}$)

Algorithm 2: CAMERA-P —Micro-Experts Pruning

Input: MoE model $\mathcal{M}$ to be pruned; calibration dataset $\mathcal{C}$; overall pruning ratio λ
Output: Pruned model $\mathcal{M}_{\text{prune}}$

1 $\mathbf{I} \leftarrow$ `Embedding`($\mathcal{C}$)
2 $\mathbf{O} \leftarrow$ `ZerosLike`($\mathbf{I}$)
3 **for** each layer $L_i \in \{L_1, L_2, \cdots, L_{N_l}\}$ **do**
4 $M \leftarrow$ `GetMoEModule`(L_i)
5 $h \leftarrow$ `RegisterForwardHook`(M)
6 $\mathbf{O} \leftarrow L_i(\mathbf{I})$
7 $(\mathbf{X}, \mathbf{Y}) \leftarrow$ `GetLayerCalibSamples`(h)
8 $\mathcal{R} \leftarrow$ CAMERA $(M, \mathbf{X}, \mathbf{Y}, \alpha)$
9 $S \leftarrow$ `TopSelect`($\mathcal{R}, 1-\lambda$)
10 $\mathbf{W}^{\text{up}} \leftarrow$ `Concat`($[\mathbf{W}_j^{\text{up}}$ for $j = 1$ to $N_E]$)
11 $\mathbf{W}^{\text{gate}} \leftarrow$ `Concat`($[\mathbf{W}_j^{\text{gate}}$ for $j = 1$ to $N_E]$)
12 $\mathbf{W}^{\text{down}} \leftarrow$ `Concat`($[\mathbf{W}_j^{\text{down}}$ for $j = 1$ to $N_E]$)
13 $\mathbf{W}^{\text{up}}, \mathbf{W}^{\text{gate}}, \mathbf{W}^{\text{down}} \leftarrow \mathbf{W}^{\text{up}}_{S,:}, \mathbf{W}^{\text{gate}}_{S,:}, \mathbf{W}^{\text{down}}_{:,S}$
14 $\mathbf{O} \leftarrow L_i(\mathbf{I})$
15 $\mathbf{I}, \mathbf{O} \leftarrow \mathbf{O}, \mathbf{I}$
16 **end for**
17 **return** $\mathcal{M}_{\text{prune}}$

4 Structured Pruning

In this section, we propose CAMERA-P, a multi-matrix joint pruning algorithm for MoE layers based on micro-expert redundancy analysis, along with the associated experiments and analytical results.

4.1 Pruning Framework CAMERA-P

We aim to prune a specified proportion λ of micro-experts in each MoE module. Leveraging the CAMERA algorithm, we identify the less important micro-experts in each layer. Specifically, for each selected micro-expert, all three associated weight vectors are simultaneously zeroed out. The detailed procedure is outlined in Algorithm 2, which applies pruning layer by layer. First, lines $5 \sim 7$ collect calibration samples from the MoE module, including the input and output hidden states. Then, the CAMERA greedy algorithm is used to rank all micro-experts and determine the set S of those to retain. Within each expert, only the micro-experts with indices in S are preserved. Finally, the post-pruned output $\mathbf{O}$ is recomputed, proceeding to the next layer.

4.2 Experimental Setup

Models and Data We evaluate our method on three modern MoE models with more, smaller experts: Deepseek-MoE-16B, Qwen2-57B-A14B, and Qwen3-30B-A3B. Each is tested under 20%-60% overall pruning ratios. Earlier, fewer-expert models are discussed in Appendix. All methods use Wikitext2 for calibration; ours uses 128 sequences of 2048 tokens. Other settings are in Appendix.

λ	Method	Wiki2	C4	BoolQ	OBQA	RTE	Wino.	Hella.	PIQA	MathQA	ARC-e	ARC-c	Avg.
Deepseek-MoE-16B-base (2 shared experts + 64 common experts, 2+top-4 experts are activated)													
0%	Original	6.51	9.05	72.45	44.00	63.54	70.24	77.37	80.68	31.52	72.89	47.86	62.28
20%	NAEE	6.77	10.07	67.83	42.40	62.09	69.53	74.63	78.34	30.99	**72.56**	**46.24**	60.51
	D^2-MoE	7.29	12.62	**69.32**	41.40	61.01	69.22	69.87	76.44	29.45	71.29	42.75	58.97
	CAMERA-P	**6.57**	**9.84**	68.01	**44.00**	**64.62**	**70.17**	**75.02**	**78.62**	**31.46**	71.80	45.56	**61.03**
40%	NAEE	8.01	12.80	62.26	39.60	57.40	63.69	66.16	75.41	27.37	64.06	38.48	54.94
	D^2-MoE	8.38	17.22	66.05	36.60	57.03	66.77	58.74	71.44	27.67	66.03	38.57	54.32
	CAMERA-P	**6.93**	**11.68**	**70.64**	**43.20**	**58.48**	**68.51**	**69.04**	**75.41**	**29.01**	**70.71**	**42.24**	**58.58**
60%	NAEE	15.47	29.44	51.65	30.60	**58.48**	53.67	47.50	65.88	22.31	48.90	28.50	45.28
	D^2-MoE	12.13	34.54	61.78	31.60	53.43	61.09	43.29	63.87	23.65	50.59	31.14	46.72
	CAMERA-P	**8.68**	**18.10**	**62.60**	**40.20**	56.32	**64.33**	**56.53**	**67.90**	**26.16**	**54.88**	**35.67**	**51.62**
Qwen2-57B-A14B (8 shared experts + 64 common experts, 8+top-8 experts are activated)													
0%	Original	5.92	8.22	86.39	44.20	75.45	73.48	82.56	81.61	38.22	69.53	49.23	66.74
20%	NAEE	6.32	8.87	86.08	43.80	74.73	73.79	81.19	**81.55**	35.14	69.36	49.32	66.11
	D^2-MoE	6.15	9.66	**87.21**	44.00	74.01	**75.37**	79.94	80.63	39.00	69.53	47.70	66.38
	CAMERA-P	**6.03**	**8.73**	85.57	**45.20**	**74.73**	74.03	**82.02**	81.18	**41.71**	**70.37**	**50.68**	**67.28**
40%	NAEE	7.72	10.56	82.20	42.00	69.68	70.88	76.68	79.81	34.04	70.20	49.83	63.92
	D^2-MoE	6.37	12.74	85.50	42.40	74.62	72.93	72.25	76.99	35.48	70.58	48.81	64.40
	CAMERA-P	**6.31**	**9.75**	**86.42**	**45.40**	**74.73**	**73.88**	**80.34**	**80.03**	**36.85**	**73.10**	**50.51**	**66.81**
60%	NAEE	16.68	21.69	66.64	34.20	56.68	62.04	57.88	70.24	25.06	55.89	33.96	51.40
	D^2-MoE	11.56	25.92	75.69	35.40	72.56	70.80	57.35	70.35	27.10	59.22	38.40	56.32
	CAMERA-P	**7.24**	**12.45**	**83.79**	**42.40**	**78.70**	**72.38**	**73.23**	**76.39**	**35.78**	**73.11**	**50.77**	**65.17**
Qwen3-30B-A3B (128 common experts, top-8 experts are activated)													
0%	Original	8.70	12.15	88.69	44.40	81.23	70.64	77.63	80.52	59.20	79.08	56.31	70.86
20%	NAEE	8.72	12.44	**88.74**	44.40	**83.39**	69.85	77.32	80.14	49.27	77.31	**56.31**	69.64
	D^2-MoE	9.68	18.52	85.90	42.80	80.87	68.19	74.26	78.40	48.81	71.46	46.50	66.35
	CAMERA-P	**8.48**	**12.25**	88.50	**44.40**	82.67	**70.56**	**77.51**	**80.30**	**52.80**	**78.33**	54.43	**69.94**
40%	NAEE	9.29	**13.87**	**87.21**	43.00	74.01	**68.74**	73.50	77.74	42.91	72.72	52.13	65.77
	D^2-MoE	20.40	35.48	85.50	42.00	76.53	64.33	69.87	71.44	40.90	69.78	44.62	62.77
	CAMERA-P	**9.29**	15.58	86.97	**43.00**	**79.42**	67.64	**73.79**	**78.51**	**44.96**	**77.31**	**55.55**	**67.46**
60%	NAEE	**12.08**	**19.37**	72.23	36.60	68.95	63.93	62.06	71.87	28.68	**66.92**	42.92	57.13
	D^2-MoE	32.13	68.36	70.64	35.40	64.62	59.12	58.21	63.23	26.00	57.37	33.57	52.02
	CAMERA-P	12.48	24.48	**82.08**	**41.00**	**68.95**	**64.64**	**64.90**	**73.01**	**30.89**	62.46	**43.35**	**59.03**

Table 1: Main pruning results of evaluation experiment on three mainstream MoE models. The best scores are in bold. We also test the performance of the original model (16-bit) as a reference.

Baselines We compare CAMERA-P with two strong baselines: NAEE (Lu et al. 2024), which prunes entire experts directly, and D^2-MoE (Gu et al. 2025), a recent strong merge-then-compress approach. To address the scalability issue with $N_E \geq 8$ in NAEE, we follow MoE-I^2 (Yang et al. 2024) and adopt efficient genetic search. CAMERA-P uses $\alpha = 0.95, 0.95, 1.00$ for the three models.

Evaluation To evaluate baseline performance, we calculate perplexity on randomly sampled sequences from Wikitext2 (Merity et al. 2017) and C4 (Raffel et al. 2020)—lower is better. We also report accuracy on nine zero-shot downstream tasks, including Winogrande (Sakaguchi et al. 2021), HellaSwag (Zellers et al. 2019), PIQA (Bisk et al. 2020), BoolQ (Clark et al. 2019), ARC-e/ARC-c (Clark et al. 2018), OBQA (Mihaylov et al. 2018), MathQA (Amini et al. 2019) and RTE (Wang et al. 2018). We prioritize outputting `acc_norm` from LM-Evaluation-Harness (Gao et al. 2024).

4.3 Main Results

We evaluate both shared and non-shared expert models, as shown in Table 1. Across all models and pruning ratios, our method consistently outperforms strong baselines in both perplexity and accuracy, especially under higher pruning rates. D^2-MoE often ranks second, except in certain Qwen3 configurations. However, its SVD-based approach suffers from numerical instability in several Qwen2/3 layers, while our method remains stable. We believe the superior performance of CAMERA-P stems from preserving the functional structure of micro-experts across matrices, maintaining important ones with full precision. Moreover, CAMERA-P is

Wbits	Method	Wiki2	C4	BoolQ	OBQA	RTE	Wino.	Hella.	PIQA	MathQA	ARC-e	ARC-c	Avg.
Deepseek-MoE-16B-base (2 shared experts + 64 common experts, 2+top-4 experts are activated)													
16-bit	Original	6.51	9.05	72.45	44.00	63.54	70.24	77.37	80.68	31.52	72.89	47.86	62.28
2.25-bit	GPTQ	11.36	14.34	61.44	37.60	55.23	64.01	64.35	75.57	25.23	62.25	35.41	53.45
	MC	11.10	16.54	60.82	38.80	**56.32**	64.25	67.77	76.33	24.69	62.46	38.65	54.45
	CAMERA-Q†	11.28	14.70	61.68	38.40	53.60	64.33	61.45	76.88	26.33	59.04	32.50	52.69
	CAMERA-Q	**9.51**	**12.26**	**66.94**	**39.00**	55.23	**66.46**	**69.21**	**77.04**	**28.07**	**66.54**	**40.52**	**56.56**
Qwen3-30B-A3B (128 common experts, top-8 experts are activated)													
16-bit	Original	8.70	12.15	88.69	44.40	81.23	70.64	77.63	80.52	59.20	79.08	56.31	70.85
2.25-bit	GPTQ	13.72	16.06	72.51	33.80	68.59	59.83	66.04	70.62	24.36	46.89	32.17	52.76
	MC	13.59	15.20	**72.57**	34.20	**68.95**	61.08	68.30	68.28	25.06	45.49	31.14	52.79
	CAMERA-Q†	14.06	15.79	71.58	34.20	67.15	62.04	67.14	71.44	24.52	45.90	28.82	52.53
	CAMERA-Q	**12.13**	**14.97**	72.32	**36.80**	64.62	**62.75**	**71.19**	**74.16**	**26.00**	**50.38**	**34.56**	**54.75**

Table 2: Main mixed-precision quantization results of evaluation experiment. The best scores are in bold.

Algorithm 3: CAMERA-Q —Mixed-precision Quantization

Input: MoE model $\mathcal{M}$ to be quantized; calibration dataset $\mathcal{C}$; ratio list $\{r_1, r_2, r_3\}$; bit-width list $\{b_1, b_2, b_3\}$
Output: Quantized model $\mathcal{M}_{\text{quant}}$

1: Get initial $\mathbf{I}, \mathbf{O}$ such as in CAMERA-P
2: **for** each layer $L_i \in \{L_1, L_2, \cdots, L_{N_l}\}$ **do**
3: Get calib-samples $(\mathbf{X}, \mathbf{Y})$ such as in CAMERA-P
4: $\mathcal{R} \leftarrow \text{CAMERA}\ (M, \mathbf{X}, \mathbf{Y}, \alpha)$
5: $S_1, S_2, S_3 \leftarrow \texttt{ListSplitByRatio}(\mathcal{R}, r_1, r_2, r_3)$
6: **for** each expert $j = 1$ to N_E **do**
7: $p, q \leftarrow j \cdot d_{\text{ff}},\ (j{+}1) \cdot d_{\text{ff}}$
8: $s_1, s_2, s_3 \leftarrow \texttt{GetSubIndex}(S_1, S_2, S_3, [p, q])$
9: **for** each bit-width $b_k \in \{b_1, b_2, b_3\}$ **do**
10: $\mathbf{W}^{\text{up}}_{s_k,:} \leftarrow \texttt{Quantize}(\mathbf{W}^{\text{up}}_{s_k,:}, b_k)$
11: $\mathbf{W}^{\text{gate}}_{s_k,:} \leftarrow \texttt{Quantize}(\mathbf{W}^{\text{gate}}_{s_k,:}, b_k)$
12: $\mathbf{W}^{\text{down}}_{:,s_k} \leftarrow \texttt{Quantize}(\mathbf{W}^{\text{down}}_{:,s_k}, b_k)$
13: **end for**
14: **end for**
15: Recompute $\mathbf{I}, \mathbf{O}$ such as in CAMERA-P
16: **end for**
17: **return** $\mathcal{M}_{\text{quant}}$

a training- and gradient-free method that completes in just 0.1 GPU hours, making it over 100x faster than competing methods. As for NAEE, this brute-force combinatorial approximation seems beneficial only at low pruning ratios.

5 Mixed-precision Quantization

In this section, we propose CAMERA-Q, a cross-expert mixed-precision bit allocation strategy for MoE models, and present the corresponding experimental results. Notably, CAMERA-Q is not introduced as a standalone quantization algorithm, but rather as a complementary component that can be **integrated** with any existing weight quantization method to enable mixed-precision quantization.

5.1 Mixed-precision Strategy CAMERA-Q

We illustrate CAMERA-Q in a setting with three predefined precision levels, where each matrix is partitioned into three segments. As shown in line 4 of Algorithm 3, CAMERA provides a global importance ranking of all micro-experts, forming the basis for mixed-precision assignment. Line 5 then divides this ranking into index sets S_i, each corresponding to a different precision level. Notably, these sets span all experts across the entire MoE layer. For each expert, we extract the indices s_i of its micro-experts within each S_i, and quantize the corresponding sub-matrices accordingly. Lines from 10 to 12 handle rows and columns differently to ensure consistent precision within each micro-expert. Additionally, micro-experts in each expert are reordered beforehand so that those assigned the same precision level are colocated.

5.2 Experimental Setup

We perform 2-bit mixed-precision quantization on MoE layers on Deepseek-MoE-16B and Qwen3-30B-A3B, focusing on comparisons with single-precision GPTQ (Frantar et al. 2022) and the recent MC method (Huang et al. 2025) that applies mixed precision at the expert level. For fairness, CAMERA-Q also adopts GPTQ for its quantization step (Lines 10∼12 in Algorithm 3). We further introduce a variant baseline, CAMERA-Q†, which applies mixed precision by slicing all matrices along the input dimension (i.e., column-wise), following the most common practice. Specifically, our method is $\mathbf{W}^{\text{up}}_{s_k,:}, \mathbf{W}^{\text{gate}}_{s_k,:}, \mathbf{W}^{\text{down}}_{:,s_k}$, while CAMERA-Q† uses $\mathbf{W}^{\text{up}}_{:,s_k}, \mathbf{W}^{\text{gate}}_{:,s_k}, \mathbf{W}^{\text{down}}_{:,s_k}$, breaking precision consistency within each micro-expert. We analyze this difference in detail in Section 6.6. Precision settings are in Appendix. All baselines are calibrated on the C4 dataset, and our method uses 128 sequences of length 2048. Evaluation follows the same setting as CAMERA-P.

5.3 Main Results

As shown in Table 2, CAMERA-Q demonstrates clear superiority over other baselines on both perplexity and accuracy. GPTQ serves as a standard baseline for 2-bit single-

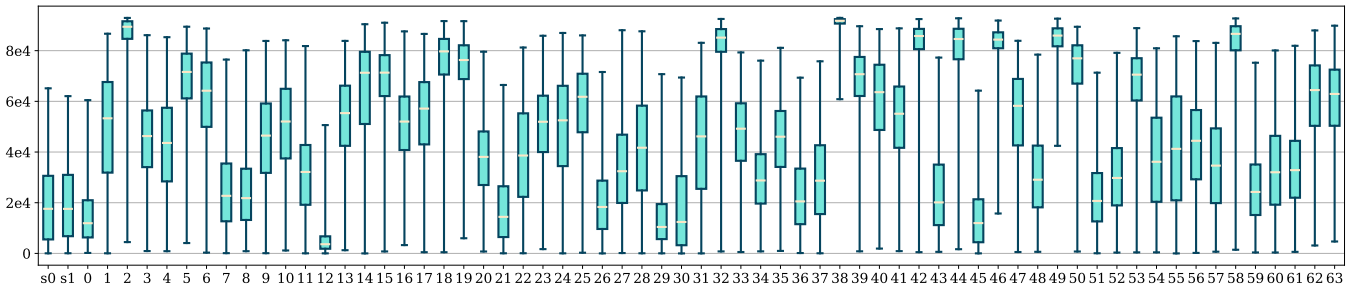

Figure 2: Distribution of micro-experts within each expert based on global ranking from CAMERA ($\lambda = 20\%$), layer 12 of Deepseek-MoE-16B. We list all 66 experts, where 'S0/S1' denotes the shared experts, and the rest are non-shared experts.

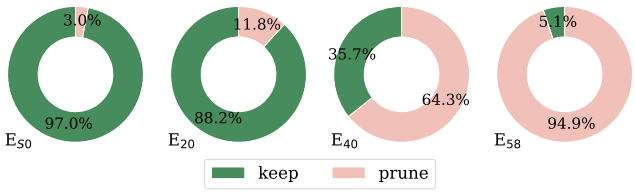

Figure 3: Pruning ratios across selected experts, taken from layer 12 of Deepseek-MoE-16B, with $\lambda = 40\%$.

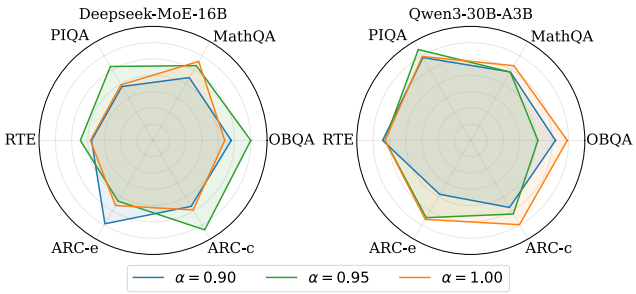

Figure 4: Task performance with varying α when $\lambda = 20\%$. The scores are scaled to highlight the differences.

precision quantization, while MC achieves slightly better results by adjusting expert-level bit allocations based on activation frequency and pre-quantization loss. However, we stress that such coarse-grained expert-level allocation fails to capture the finer differences among micro-experts. CAMERA-Q addresses this limitation, and its strong performance supports the effectiveness of a more fine-grained approach. Moreover, CAMERA-Q adopts a matrix partitioning distinct from that used in previous methods, which also contributes to its performance gains, as evidenced by the performance drop observed in CAMERA-Q†.

6 Discussion

6.1 Micro-Expert Distribution

Figure 2 illustrates the significant variation in the distribution of micro-experts across different experts within the same layer. A lower rank indicates higher importance. Notably, the two shared experts, along with experts 0, 12, 29, and 45, stand out as particularly important, as most of their micro-experts rank highly at the global level. This provides strong evidence supporting the core assumption of CAMERA and underscores the advantage of fine-grained expert pruning over coarse-grained methods, as shown in Figure 3.

$\lvert\mathcal{C}\rvert$	8	16	32	64	128	256
Wikitext2	6.66	6.60	6.59	6.57	6.57	6.56
C4	7.28	7.13	7.07	6.94	6.95	6.92

Table 3: Perplexity on Wikitext2 with different sizes and sources of calibration data. $\lvert\mathcal{C}\rvert$ denotes the size of $\mathcal{C}$.

Method	NAEE	D^2-MoE	CAMERA-P
$\lambda = 20\%$	1.00x	1.03x	1.06x
$\lambda = 40\%$	1.00x	1.04x	1.42x
$\lambda = 60\%$	1.00x	1.08x	1.48x

Table 4: Decoding speed comparison of pruned MoE layer in Deepseek-MoE-16B. The batch size is 64 tokens.

6.2 Approximation Errors

Our better performance is interpretable. By retaining the most key micro-experts, the outputs of the pruned MoE layers under CAMERA-P remain closer to those of the original model—both in terms of L2 distance and cosine similarity. Please refer to Appendix for detailed results.

6.3 Ablation on Balance Coefficient

The α in Equation 11 slightly affects the ranking of micro-experts and downstream performance, but has little impact on perplexity or average accuracy. Figure 4 shows how different α influences task scores. We infer that task- or domain-specific micro-experts lead to this behavior. In experiments, we choose the α that produced the highest scores.

6.4 Ablation on Calibration Dataset

CAMERA achieves strong performance using only a small amount of general-domain data. Table 3 shows that CAMERA-P is insensitive to both the source and size of calibration data when pruning 20% of Deepseek-MoE-16B.

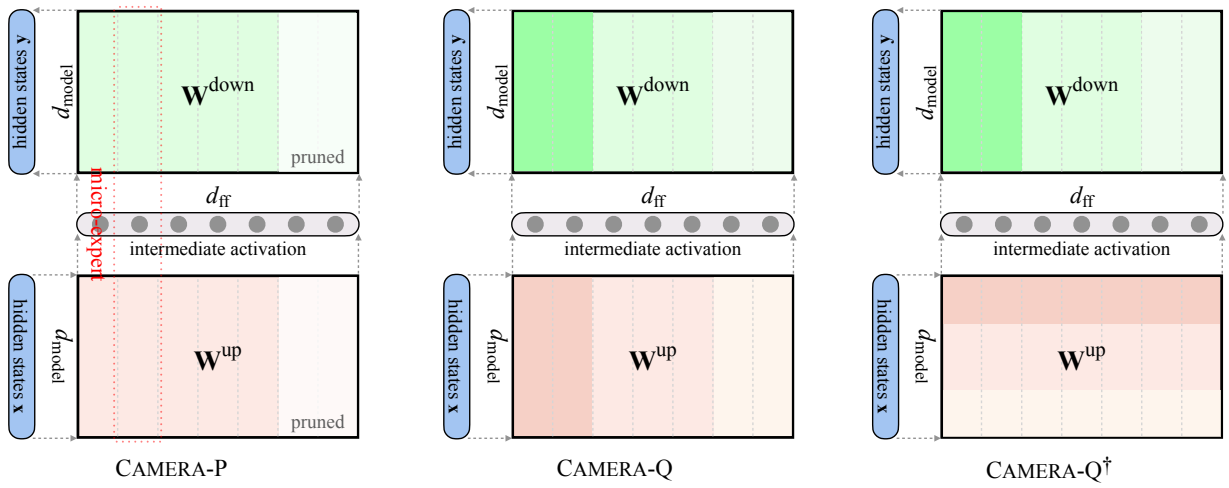

Figure 5: Matrix calculation flow of CAMERA-P, CAMERA-Q and CAMERA-Q†. For simplicity, we omit the matrix $\mathbf{W}^{\text{gate}}$. The red dashed box on the left indicates the weight of a micro-expert. In CAMERA-Q and CAMERA-Q†, we use light and dark colors to indicate the lower and higher bit-width of the weights.

Method	Wiki2	C4	BoolQ	OBQA	RTE	Wino.	Hella.	PIQA	MathQA	ARC-e	ARC-c	Avg.
openPangu Embedded-7B-V1.1 (dense model, only 1 expert)												
Original (16-bit)	34.94	43.73	77.74	29.80	67.87	55.25	59.98	68.72	26.20	54.97	35.49	52.89
CAMERA-P (20%)	44.35	48.67	75.20	31.00	65.34	54.14	54.79	66.43	23.75	52.90	35.41	51.00
+Wanda (20%+20%)	42.96	48.11	74.56	30.80	66.06	55.09	54.34	67.19	24.09	53.07	34.81	51.11

Table 5: Results of CAMERA-P and integrating Wanda. "20%+20%" denotes a pruning ratio of "20%+80%×20%=36%".

6.5 Decoding Speed on Pruned MoE Layer

NAEE does not reduce weights during decoding, and while D^2-MoE performs rank reduction, it requires additional steps. In contrast, CAMERA-P directly reduces weights, enabling more efficient decoding, as shown in Table 4.

6.6 Integrity of Micro-Experts

A central insight of our method is that jointly compressing multiple matrices preserves the functional integrity of micro-experts. Figure 5 compares three different methods. All share the same FFN structure: the input $\mathbf{x}$ is transformed by $\mathbf{W}^{\text{up}}$ and $\mathbf{W}^{\text{gate}}$ into intermediate activations, followed by $\mathbf{W}^{\text{down}}$ to produce $\mathbf{y}$. As highlighted in the red dashed box, each row of $\mathbf{W}^{\text{up}}$ maps uniquely to a column of $\mathbf{W}^{\text{down}}$, forming a one-to-one micro-expert. For visualization, $\mathbf{W}^{\text{up}}$ is shown transposed and micro-experts are energy-ranked from left to right.

CAMERA-P prunes low-energy micro-experts while keeping the remains intact. CAMERA-Q assigns higher precision to higher-energy micro-experts, enforcing uniform precision within each. In contrast, CAMERA-Q† slices $\mathbf{W}^{\text{up}}$ and $\mathbf{W}^{\text{gate}}$ along an orthogonal dimension, quantizing each weight with multiple precisions. Like many existing mixed-precision approaches, CAMERA-Q† operates on individual matrices and allocates bits along the input dimension (e.g., using $\mathbf{H} = \mathbf{X}\mathbf{X}^{\text{T}}$), overlooking cross-matrix expert structure. This often gives high-energy micro-experts inconsistent or insufficient precision. Our experiments show that enforcing precision consistency within each micro-expert, as in CAMERA-Q, yields substantially better performance.

6.7 Application on Dense Model

CAMERA investigates redundancy that spans across matrices within MoE layers. After applying cross-matrix compression, the resulting MoE layers can still be further compressed using single-matrix methods such as Wanda (Sun et al. 2024). Moreover, our approach can also be directly applied to structured pruning or mixed-precision quantization in dense-architecture models. As an illustrative example, we apply CAMERA-P to prune 20% of the FFN layers in the *openPangu-7B* model (Chen et al. 2025) running on *Ascend 910B*. On top of this, we further apply Wanda to perform an additional 20% unstructured pruning. The resulting performance is summarized in Table 5. These results demonstrate that expert-level pruning and intra-matrix pruning can be seamlessly combined to achieve higher overall pruning ratios while maintaining lossless performance.

7 Conclusion

We propose CAMERA-P, a novel, effective, and efficient MoE pruning method grounded in a cross-matrix perspective of micro-experts and guided by the CAMERA ranking algorithm. We further highlight the importance of micro-expert-oriented mixed-precision idea in CAMERA-Q.

Acknowledgements

We gratefully acknowledge the support of the National Natural Science Foundation of China (NSFC) via grants 62236004, 62206078 and 62476073. Additional gratitude is extended to our collaborators and colleagues for their insightful discussions during the development of this work.

References

Ai, M.; Wei, T.; Chen, Y.; Zeng, Z.; Zhao, R.; Varatkar, G.; Rouhani, B. D.; Tang, X.; Tong, H.; and He, J. 2025. ResMoE: Space-efficient Compression of Mixture of Experts LLMs via Residual Restoration. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1–12.

Amini, A.; Gabriel, S.; Lin, S.; Koncel-Kedziorski, R.; Choi, Y.; and Hajishirzi, H. 2019. MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2357–2367.

Bisk, Y.; Zellers, R.; Gao, J.; Choi, Y.; et al. 2020. PIQA: Reasoning about Physical Commonsense in Natural Language. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, 7432–7439.

Chen, H.; Wang, Y.; Han, K.; Li, D.; Li, L.; Bi, Z.; Li, J.; Wang, H.; Mi, F.; Zhu, M.; et al. 2025. Pangu Embedded: An Efficient Dual-system LLM Reasoner with Metacognition. *arXiv preprint arXiv:2505.22375*.

Chen, I.; Liu, H.-S.; Sun, W.-F.; Chao, C.-H.; Hsu, Y.-C.; Lee, C.-Y.; et al. 2024. Retraining-Free Merging of Sparse Mixture-of-Experts via Hierarchical Clustering. *arXiv preprint arXiv:2410.08589*.

Clark, C.; Lee, K.; Chang, M.-W.; Kwiatkowski, T.; Collins, M.; and Toutanova, K. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2924–2936.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think You Have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *arXiv preprint arXiv:1803.05457v1*.

Duanmu, H.; Li, X.; Yuan, Z.; Zheng, S.; Duan, J.; Zhang, X.; and Lin, D. 2025. MxMoE: Mixed-precision Quantization for MoE with Accuracy and Performance Co-Design. In *Proceedings of International Conference on Machine Learning (ICML)*, 14793–14806.

Fedus, W.; Zoph, B.; and Shazeer, N. 2022. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *Journal of Machine Learning Research*, 23(120): 1–39.

Frantar, E.; and Alistarh, D. 2023. SparseGPT: Massive Language Models Can Be Accurately Pruned in One-shot. In *Proceedings of International Conference on Machine Learning (ICML)*, 10323–10337.

Frantar, E.; Ashkboos, S.; Hoefler, T.; and Alistarh, D. 2022. GPTQ: Accurate Post-training Quantization for Generative Pre-trained Transformers. *arXiv preprint arXiv:2210.17323v2*.

Gao, L.; Tow, J.; Abbasi, B.; Biderman, S.; Black, S.; DiPofi, A.; Foster, C.; Golding, L.; Hsu, J.; Le Noac'h, A.; Li, H.; McDonell, K.; Muennighoff, N.; Ociepa, C.; Phang, J.; Reynolds, L.; Schoelkopf, H.; Skowron, A.; Sutawika, L.; Tang, E.; Thite, A.; Wang, B.; Wang, K.; and Zou, A. 2024. The Language Model Evaluation Harness.

Gu, H.; Li, W.; Li, L.; Qiyuan, Z.; Lee, M. G.; Sun, S.; Xue, W.; and Guo, Y. 2025. Delta Decompression for MoE-based LLMs Compression. In *Proceedings of International Conference on Machine Learning (ICML)*, 20497–20514.

Huang, W.; Liao, Y.; Liu, J.; He, R.; Tan, H.; Zhang, S.; Li, H.; Liu, S.; and QI, X. 2025. Mixture Compressor for Mixture-of-Experts LLMs Gains More. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)*.

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

Li, L.; Qiyuan, Z.; Wang, J.; Li, W.; Gu, H.; Han, S.; and Guo, Y. 2025a. Sub-MoE: Efficient Mixture-of-Expert LLMs Compression via Subspace Expert Merging. *arXiv preprint arXiv:2506.23266*.

Li, P.; Zhang, Z.; Yadav, P.; Sung, Y.-L.; Cheng, Y.; Bansal, M.; and Chen, T. 2024. Merge, Then Compress: Demystify Efficient SMoE with Hints from Its Routing Policy. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.

Li, W.; Li, L.; Gu, H.; Huang, Y.-L.; Lee, M. G.; Sun, S.; Xue, W.; and Guo, Y. 2025b. MoE-SVD: Structured Mixture-of-Experts LLMs Compression via Singular Value Decomposition. In *Proceedings of International Conference on Machine Learning (ICML)*, 35209–35230.

Liu, A.; Feng, B.; Wang, B.; Wang, B.; Liu, B.; Zhao, C.; Dengr, C.; Ruan, C.; Dai, D.; Guo, D.; et al. 2024a. Deepseek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. *arXiv preprint arXiv:2405.04434*.

Liu, E.; Zhu, J.; Lin, Z.; Ning, X.; Blaschko, M. B.; Yan, S.; Dai, G.; Yang, H.; and Wang, Y. 2024b. Efficient Expert Pruning for Sparse Mixture-of-Experts Language Models: Enhancing Performance and Reducing Inference Costs. *arXiv preprint arXiv:2407.00945*.

Liu, Z.; Zhao, C.; Fedorov, I.; Soran, B.; Choudhary, D.; Krishnamoorthi, R.; Chandra, V.; Tian, Y.; and Blankevoort, T. 2025. SpinQuant: LLM Quantization with Learned Rotations. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)*.

Lu, X.; Liu, Q.; Xu, Y.; Zhou, A.; Huang, S.; Zhang, B.; Yan, J.; and Li, H. 2024. Not All Experts are Equal: Efficient Expert Pruning and Skipping for Mixture-of-Experts Large

Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 6159–6172.

Mahoney, M. W.; and Drineas, P. 2009. CUR Matrix Decompositions for Improved Data Analysis. *Proceedings of the National Academy of Sciences*, 106(3): 697–702.

Merity, S.; Xiong, C.; Bradbury, J.; and Socher, R. 2017. Pointer Sentinel Mixture Models. In *Proceedings of the Fifth International Conference on Learning Representations (ICLR)*.

Mihaylov, T.; Clark, P.; Khot, T.; and Sabharwal, A. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2381–2391.

Ordozgoiti, B.; Canaval, S. G.; and Mozo, A. 2018. Iterative Column Subset Selection. *Knowledge and Information Systems*, 54(1): 65–94.

Ping, B.; Wang, S.; Wang, H.; Han, X.; Xu, Y.; Yan, Y.; Chen, Y.; Chang, B.; Liu, Z.; and Sun, M. 2024. Delta-CoMe: Training-free Delta-compression with Mixed-precision for Large Language Models. *Advances in Neural Information Processing System (NeurIPS)*, 37: 31056–31077.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-text Transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.

Sakaguchi, K.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2021. Winogrande: An Adversarial Winograd Schema Challenge at Scale. *Communications of the ACM*, 64(9): 99–106.

Shao, W.; Chen, M.; Zhang, Z.; Xu, P.; Zhao, L.; Li, Z.; Zhang, K.; Gao, P.; Qiao, Y.; and Luo, P. 2024. OmniQuant: Omnidirectionally Calibrated Quantization for Large Language Models. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.

Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; and Dean, J. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *Proceedings of the Fifth International Conference on Learning Representations (ICLR)*.

Shitov, Y. 2021. Column Subset Selection is NP-complete. *Linear Algebra and its Applications*, 610: 52–58.

Sun, M.; Liu, Z.; Bair, A.; and Kolter, J. Z. 2024. A Simple and Effective Pruning Approach for Large Language Models. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.

Team, K.; Bai, Y.; Bao, Y.; Chen, G.; Chen, J.; Chen, N.; Chen, R.; Chen, Y.; Chen, Y.; Chen, Y.; et al. 2025. Kimi K2: Open Agentic Intelligence. *arXiv preprint arXiv:2507.20534*.

Tseng, A.; Chee, J.; Sun, Q.; Kuleshov, V.; and De Sa, C. 2024. QuIP#: Even Better LLM Quantization with Hadamard Incoherence and Lattice Codebooks. In *Proceedings of International Conference on Machine Learning (ICML)*, 48630–48656.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 353–355.

Xiao, G.; Lin, J.; Seznec, M.; Wu, H.; Demouth, J.; and Han, S. 2023. SmoothQuant: Accurate and Efficient Post-training Quantization for Large Language Models. In *Proceedings of International Conference on Machine Learning (ICML)*, 38087–38099.

Xie, Y.; Zhang, Z.; Zhou, D.; Xie, C.; Song, Z.; Liu, X.; Wang, Y.; Lin, X.; and Xu, A. 2024. MoE-Pruner: Pruning Mixture-of-Experts Large Language Model Using the Hints from Its Router. *arXiv preprint arXiv:2410.12013*.

Xie, Z.; Ma, Y.; Zheng, X.; Chao, F.; Sui, W.; Li, Y.; Li, S.; and Ji, R. 2025. Automated Fine-Grained Mixture-of-Experts Quantization. In *Findings of the Association for Computational Linguistics (ACL Findings)*, 27024–27037.

Xu, Y.; Han, X.; Yang, Z.; Wang, S.; Zhu, Q.; Liu, Z.; Liu, W.; and Che, W. 2024. OneBit: Towards Extremely Low-bit Large Language Models. *Advances in Neural Information Processing System (NeurIPS)*, 37: 66357–66382.

Xu, Y.; Ji, S.; Zhu, Q.; and Che, W. 2025. CRVQ: Channel-Relaxed Vector Quantization for Extreme Compression of LLMs. *Transactions of the Association for Computational Linguistics (TACL)*, 13: 1488–1506.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*.

Yang, C.; Sui, Y.; Xiao, J.; Huang, L.; Gong, Y.; Duan, Y.; Jia, W.; Yin, M.; Cheng, Y.; and Yuan, B. 2024. MoE-I^2: Compressing Mixture of Experts Models through Inter-Expert Pruning and Intra-Expert Low-Rank Decomposition. In *Findings of the Association for Computational Linguistics (EMNLP Findings)*, 10456–10466.

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. Hellaswag: Can a Machine Really Finish Your Sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 4791–4800.

Zhang, Z.; Liu, X.; Cheng, H.; Xu, C.; and Gao, J. 2024. Diversifying the Expert Knowledge for Task-Agnostic Pruning in Sparse Mixture-of-Experts. In *Findings of the Association for Computational Linguistics (ACL Findings)*, 86–102.

Zhou, Z.; Ning, X.; Hong, K.; Fu, T.; Xu, J.; Li, S.; Lou, Y.; Wang, L.; Yuan, Z.; Li, X.; et al. 2024. A Survey on Efficient Inference for Large Language Models. *arXiv preprint arXiv:2404.14294*.