

DynaMo: Runtime Switchable Quantization for MoE with Cross-Dataset Adaptation

Zihao Zheng

School of Computer Science
Peking University
Beijing, China
zhengzihao@stu.pku.edu.cn

Xiuping Cui

School of Computer Science
Peking University
Beijing, China

Size Zheng

School of Computer Science
Peking University
Beijing, China

Maoliang Li

School of Computer Science
Peking University
Beijing, China

Jiayu Chen

School of Computer Science
Peking University
Beijing, China

Yun Liang

School of Integrated Circuits
Peking University
Beijing, China

Xiang Chen[†]

School of Computer Science
Peking University
Beijing, China
xiang.chen@pku.edu.cn

Abstract—As the Mix-of-Experts (MoE) architecture increases the number of parameters in large models, there is an even greater need for model quantization. However, existing quantization methods overlook the expert dynamics of MoE across multiple datasets. Moreover, the existing static quantization cannot adapt MoE to various data change scenarios. In this paper, we perform a multi-level analysis to reveal MoE dynamics and define the significance of each channel/expert. Based on the analysis results, we propose *DynaMo*, an end-to-end MoE quantization framework. *DynaMo* adopts an expert-level mixed-precision baseline quantization strategy, which ensures the quantized MoEs are compatible with multiple existing datasets. Furthermore, *DynaMo* incorporates a channel-level dynamic switching mechanism to adapt these quantized MoE models to novel datasets. Experiments show that *DynaMo* achieves a 2.78~4.54 PPL decrease and a 1.85%~3.77% accuracy improvement in various datasets, with ~3× inference speedup and negligible overhead.

Index Terms—Mix-of-Experts, Model Quantization, Multi-Level Analysis, Cross-Dataset Adaptation

I. INTRODUCTION

As Artificial Intelligence advances into the era of LLMs, their applications have expanded exponentially. Since LLMs’ parameter density fails to adapt to the increasingly large and diverse datasets for their processing, the Mix-of-Experts (MoEs) architecture has become a promising solution [1], [2]. Each layer in MoE comprises multiple “expert” models. Trained to adapt to diverse data, these experts enable MoE to fit a wide range of data with strong performance. In inference scenarios, MoE dynamically selects and sparsely activates a subset of these experts to fit the corresponding datasets. [3]–[5].

Despite their superior parameter scalability and memory efficiency via sparse expert activation, MoEs still require model compression methods [6]–[8]. Among existing model compression methods, quantization is most effective: it reduces model size and accelerates computations by using low-precision parameter representations [9].

With advances in quantization techniques, methodology has shifted from parameter formats to the mapping between weights and complex input datasets. Methods like GPTQ [10] and

QuantEase [11] use dataset analysis to establish data-weight mappings and compensate weights during quantization. Subsequent approaches such as SmoothQuant [12] and AWQ [13] further analyze outliers in specific datasets and transfer their scales to weights via data-weight mappings to minimize quantization loss. Later methods including RPTQ [14] and Atom [15] leverage these mappings for finer-grained segmentation or reordering, then apply mixed-precision quantization to optimize the balance between compression rate and quantization loss.

However, these strategies, designed for dense LLMs, yield suboptimal performance when applied to MoEs, for the following reasons: (1) MoE architecture modifies data-weight mappings: as shown in Fig. 1 (a), each token embedding is dispatched to multiple channel weights across different experts, resulting in one-to-many correlations. By contrast, quantization methods designed for large language models (LLMs) only account for one-to-one correlations, failing to capture the complex data-weight mappings inherent to MoEs. (2) MoE has inherent expert/weight dynamics: as shown in Fig. 1 (b), MoE adapts to a wide variety of datasets through a flexible combination of experts, resulting in scalable parameters and improved performance compared to LLM. Existing quantization methods are usually calibrated on one dataset, ignoring the dynamics of MoEs’ expert/weight dynamics.

Recent studies (e.g., MoQE [16], MoEPTQ [17]) have explored extending dense LLM quantization to MoEs, but they overlook the aforementioned characteristics of MoEs. Moreover, the quantified MoE derived from these strategies lacks the adaptability to multi-data scenarios, a characteristic that contradicts the original design intention of MoE. Therefore, it is crucial to conduct a thorough analysis of the complex mappings and intrinsic dynamics of MoEs, and to further rethink the quantization methods for MoE across multiple datasets.

In this paper, we conduct a multi-level analysis of MoEs’ data-weight mappings and expert/weight dynamics, building metrics to capture the dynamics and expert/weight significance. Based on the analysis results, we propose an end-to-end MoE

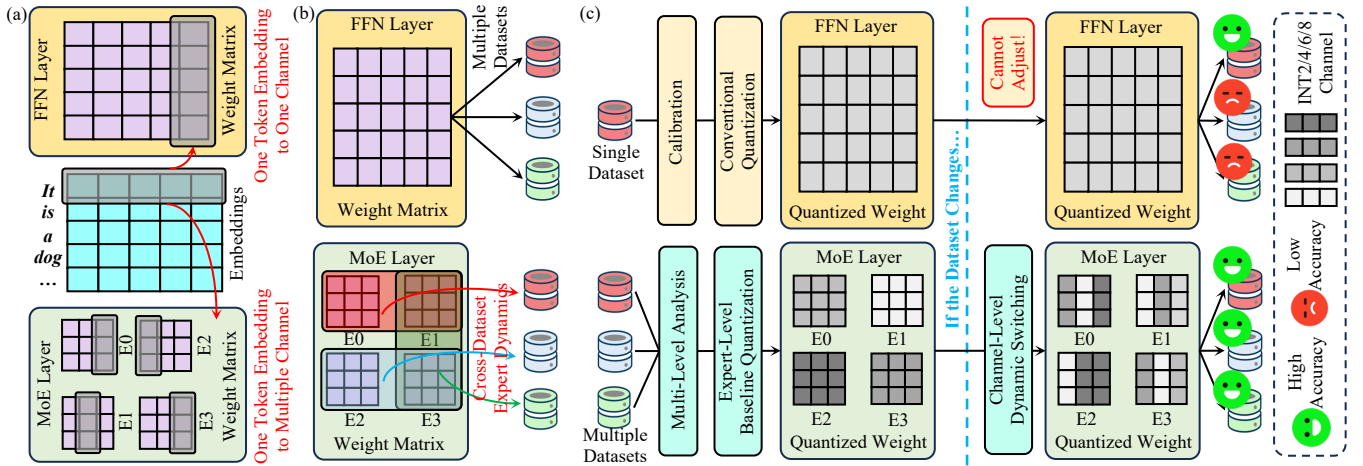


Fig. 1: (a) MoE’s One-to-Many Data-Weight Mappings (b) MoE’s Expert/Weight Dynamics across Multiple Datasets (c) A Brief Overview of the Proposed *DynaMo* Quantization Framework

quantization framework called *DynaMo*, as shown in Fig. 1 (c). Specifically, *DynaMo* adopts an expert-level mixed-precision baseline quantization strategy, which ensures the quantized MoEs are compatible with multiple existing datasets. Furthermore, *DynaMo* incorporates a channel-level dynamic switching mechanism to adapt these quantized MoE models to novel datasets. Overall, our contributions are as follows:

- We propose a multi-level analysis method for MoEs, quantitatively defining expert/weight significance within a single dataset and profiling its cross-dataset dynamics.
- Based on the analysis results, we propose *DynaMo*, an end-to-end quantization framework for MoEs, which contains both expert-level and channel-level strategies, ensuring the performance across multiple datasets.
- We implement *DynaMo* based on existing hardware and software, and conducted targeted optimizations. We further compare *DynaMo* with SOTA quantization methods, analyzed its scalability and associated overhead in detail.

Experimental results show that, compared to SOTA methods (i.e., GPTQ [10] and MoEPTQ [17]), *DynaMo* achieves lower PPL (2.78~4.54) and higher accuracy (1.85%~3.77%) on various tasks. Overall, the proposed *DynaMo* delivers superior quantization performance with minimal overhead and suits diverse datasets and MoEs.

II. PRELIMINARY

A. MoE Model

Architecture and Mechanism. As illustrated in Fig. 1 (a), dense LLMs are structured as a series of interconnected blocks. While in MoEs, the Feed-Forward Network (FFN) is replaced with multiple expert models [2], [18]. Conventional expert selection depends on the Top-K mechanism, which selects a fixed number of experts per layer [1], [19]. Some studies also adopt soft or dynamic expert utilization strategies, activating different numbers of experts according to the characteristics of various data [20]–[22].

Data-Weight Mappings. MoEs’ architecture and mechanism modifies data-weight mappings. Unlike LLMs, which use a

single FFN to establish one-to-one mappings between channel weights and individual token embeddings, MoEs employ multiple experts. This design results in each token embedding corresponding to multiple channels across different experts.

B. Model Quantization

Dense LLM Quantization. Since the rise of LLMs, quantization methods have shifted focus from parameter formats to dataset analysis to minimize accuracy loss. GPTQ [10] and QuantEase [11] establish data-weight mappings to compensate weights during quantization, curbing accuracy drops. SmoothQuant [12] and AWQ [13] target data outliers, leveraging such mappings for numerical smoothing or precision mixing to facilitate quantization. RPTQ [14] and Atom [15] further segment and reorder weight matrices based on data-weight correlations, developing channel-wise mixed-precision quantization for enhanced performance. These dataset-aware quantization methods achieves good performance.

Challenge of MoE Quantization. In MoE, the expert selection process is inherently dataset-dependent, leading to complex data-weight mappings and dynamic expert/weight dynamics across different datasets. This dynamic behavior presents a fundamental mismatch with static quantization schemes, which typically assume a fixed dataset. As a result, applying statically quantized MoE models to new datasets often leads to degraded accuracy due to MoE’s dynamics.

III. MULTI-LEVEL MOE DYNAMICS ANALYSIS

During MoE inference, each token’s parsing activates multiple experts’ channel weights, which makes it challenging to reveal the specific significance of each channel in each expert. Moreover, MoE’s token-expert mappings cover both expert-level dispatching and intra-expert channel activations, making it hard to evaluate expert importance solely through token routing frequency or isolated activation statistics. Thus, we propose a multi-level analysis to characterize MoE dynamics and better assess the significance of each channel and expert.

A. Dataset-Specific Expert Weight Significance

Since each token in the dataset is assigned to multiple experts in MoE, the token utilization of MoE differs from that of

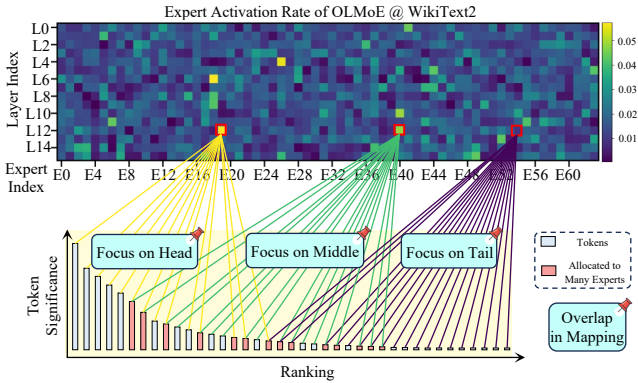


Fig. 2: Data-Weight Mappings of MoEs

LLMs. Thus, for a given dataset, we calculate the utilization rate of each token, and then establish a dataset-specific token ranking, as shown at the bottom of Fig. 2. Based on this token ranking, we randomly select tokens from the provided dataset and perform MoE inference until the entire dataset is traversed. During this process, we use a hook function to get the output logits of each channel weight in MoEs. We perform a fine-grained significance analysis regarding each channel weight in each expert, namely, the mapping correlation to each token. After that, we conducted a statistical analysis of the mapping to summarize the significance of each channel weight corresponding to the tokens in the given dataset.

B. Cross-Dataset Expert Significance Dynamics

By analyzing the significance of channel weights in each expert based on token utilization, we can determine the importance of each expert model under a single dataset. However, the importance of each expert depends on dataset-specific token utilization, and this utilization changes as the dataset varies. In such cases, traditional quantization methods, which rely on calibration with a single dataset, cannot ensure universality. Instead, these methods require tracking dataset changes to perform recalibration and quantization, a process that introduces substantial overhead. Therefore, we aim to develop a relatively general method for quantifying dynamically variable MoE models. To achieve this goal, we need to explore how experts behave dynamically as the dataset changes.

Based on the significance of channel weights, we propose “expert significance” to quantify and reflect the importance of each expert within a single dataset. Specifically, assume each layer in MoE contains N experts. The given dataset has T tokens. \mathcal{W}_{ch} means the channel weight. And $\mathcal{S}_{\text{ch}}(\cdot)$ are the significance of channel weights. Expert significance can be computed via Eq. (1).

$$\left\{ \mathcal{S}_{\text{exp}}^j = \frac{\sum_{i=1}^t \left\{ \mathcal{S}_{\text{ch}} \left(\sum_{k=1}^K \mathcal{W}_{\text{ch}}^k \mid \text{token} = \tau^i \right) \right\}}{\sum_{i=1}^T \left\{ \mathcal{S}_{\text{ch}} \left(\sum_{k=1}^K \mathcal{W}_{\text{ch}}^k \mid \text{token} = \tau^i \right) \right\}} \right\}_{j=1}^N, \quad (1)$$

Subsequently, we calculate the expert significance of the same MoE layer across different datasets (WikiText2 [23] and C4 [24]). The results are shown in Fig. 3). Under WikiText2 dataset, the 28th expert exhibits high significance (>0.5), while under C4 dataset, its significance is very low (<0.1). The

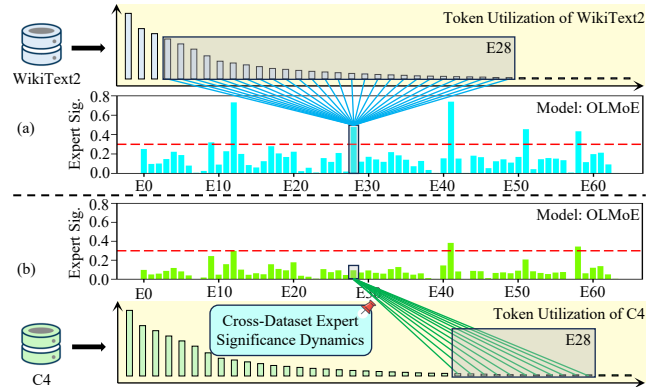


Fig. 3: Cross-Dataset Expert Significance Dynamics

proposed expert significance metric effectively reveals the dynamics of experts in MoEs across different datasets.

C. Synthesized Expert Baseline Significance

with Cross-Dataset Dynamics

The analysis presented above has elucidated the dynamics of expert significance across different datasets; however, these dynamics remain isolated from one another. Specifically, these dynamics exhibit a strong correlation with individual datasets and cannot be unified across different data sources. To adapt MoE to multiple datasets, developing an index capable of characterizing an expert’s synthesized performance across these diverse datasets is essential. Therefore, we collect the expert significance under various datasets and fit them to a joint distribution, as Eq. 2 shows. \mathcal{J} means the joint distribution. \mathcal{D} means the dataset, and $\{\mathcal{S}_{\text{exp}}^j\}_{j=1}^N \parallel \mathcal{D}_1$ means the expert significance under the first dataset.

$$\mathcal{J} \leftarrow \text{Fit} \left(\left\{ \mathcal{S}_{\text{exp}}^j \right\}_{j=1}^N \parallel \mathcal{D}_1, \dots, \left\{ \mathcal{S}_{\text{exp}}^j \right\}_{j=1}^N \parallel \mathcal{D}_n \right). \quad (2)$$

Fig. 4 illustrates the results of the joint distribution. The diagonal bars in this joint distribution represent the synthesized expert significance across multiple datasets, whereas the remaining bars depict the dynamics of expert significance. This method yields a straightforward synthesized indicator for evaluating expert dynamics across multiple datasets, which can be directly applied to our quantization design.

IV. DATASET-AWARE MOE QUANTIZATION

This section introduces a dataset-aware MoE quantization framework, called *DynaMo*. It first performs expert-level

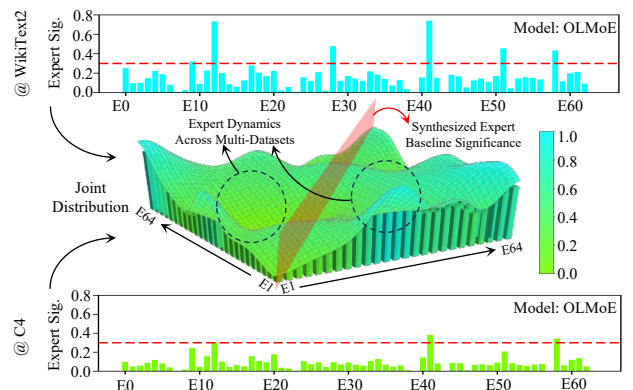


Fig. 4: Joint Distribution of the Expert Significance

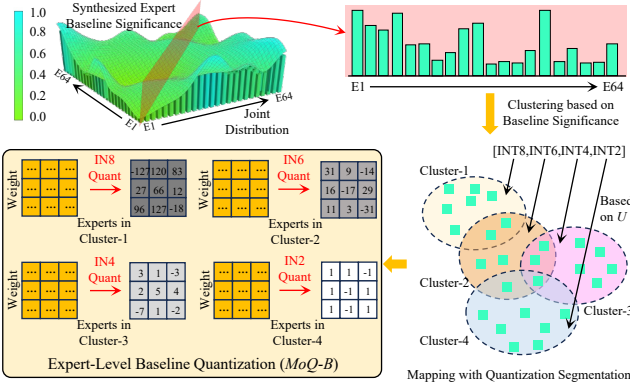


Fig. 5: Expert-level Mix-Precision Baseline Quantization

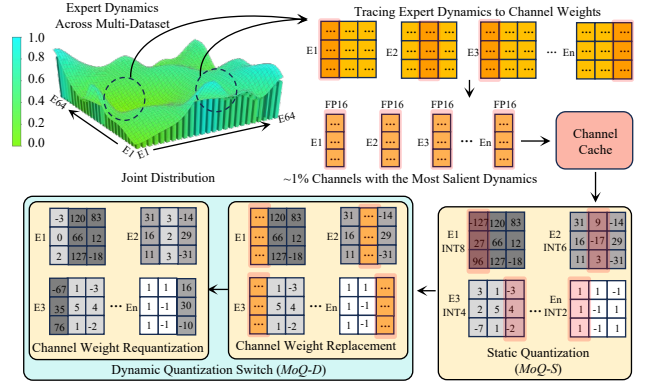


Fig. 6: Channel-level Dynamic Quantization Switching

mixed-precision static quantization (strategy determined by synthesized expert significance across datasets), then maps expert dynamics to channels and incorporates a fine-grained quantization switching mechanism.

A. Expert-Level Baseline Quantization

Due to the software and hardware limitations, existing quantization precision is mainly focused on four types (INT2, INT4, INT6, and INT8) [9]. Although we can simply partition the synthesized baseline significance to four types, in fact, different baseline significance distributions still require us to accurately delineate the boundaries.

Expert Baseline Significance Mapping with Quantization Segmentation. Based on the synthesized expert baseline significance, we cluster the experts into four categories using the proposed Alg. 1. We compute the number of tokens corresponding to each expert, and then initialize the clustering score (U) as the proportion of these expert-specific tokens relative to the total number of tokens in the dataset. Subsequently, we iteratively update the cluster centers by leveraging the product of dataset correlation and expert baseline significance. We employ the cluster center V and clustering score U as evaluation metrics to determine the quantization precision for each cluster. Specifically, the cluster with the highest U -value corresponds to the experts of greatest significance and is thus assigned the highest precision (INT8). Conversely, the cluster with the lowest U -value is assigned the lowest precision (INT2).

Quantization for Expert Located in Clustering Boundary. As illustrated in Fig. 5, the clustering results exhibit overlap across different categories. For experts situated in these over-

lapping regions, we adopt a low-precision-first strategy. The strategy is supported by experimental results, which demonstrate that the precision of these experts exerts a negligible impact on the loss of quantization accuracy. Applying low-bit quantization to these experts effectively increases the overall compression ratio. Therefore, experts in the overlapping regions are assigned to the low-precision cluster for quantization.

After determining the quantization precision for each cluster, we perform mix-precision quantization at the expert level. Given computational complexity considerations, we adopt uniform quantization to reduce the computational overhead of the dequantization process. It is important to note that all aforementioned analyses, along with the baseline quantization process, are conducted offline and do not compromise the inference speed of the quantized MoE model.

B. Channel-Level Dynamic Quantization Switching

In the preceding section, we discussed baseline precision determination and the adjustment of key boundary experts during quantization across multiple datasets. However, applying dynamic switching to all expert parameters would lead to prohibitive optimization costs. As analyzed in Section III, MoE dynamics originate from the mapping between channel weights and tokens. Tracing these dynamics to individual channel weights, we find only $\sim 1\%$ of channels in each expert exhibit the most prominent dynamics. This observation is also supported by existing research [13]. Dynamic channel switching in response to dataset changes allows the model to attain near-optimal quantization performance with minimal computational overhead. Yet, this requires addressing two key aspects: channel replacement and precision redetermination.

Channel Weights Replacement based on Caching. The model undergoing baseline quantization suffers from irrecoverable quantization loss across each channel. Consequently, we are unable to restore the quantized channel weights to a higher precision; we can only adjust them to a lower precision. To enable flexible adjustment of channel weight precision, the original unquantized FP16 channels must be retained. As shown in the upper right part of Fig. 6, we cache $\sim 1\%$ of the channels with the most salient dynamics. At the same time, to facilitate rapid searching, we also cache their channel indexes and the indexes of the corresponding experts. As shown in the lower right part of Fig. 6, these cached indexes allow fast and accurate

Algorithm 1 Clustering Algorithm based on Synthesized Expert Baseline Significance

Require: Synthesized expert baseline significance $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$,
Number of clusters $c = 4$, Fuzzifier $m > 1$, Tolerance ϵ
Ensure: Cluster centers $\mathbf{V} = \{v_1, v_2, \dots, v_c\}$, Cluster scores \mathbf{U}
1: Initialize \mathbf{U} with $u_{ik} \in [0, 1]$ and $\sum_{k=1}^c u_{ik} = 1$ for all i
2: **repeat**
3: **for** $k = 1, \dots, c$ **do** $v_k \leftarrow \sum_{i=1}^n u_{ik}^m x_i / \sum_{i=1}^n u_{ik}^m$
4: **end for**
5: **for** $\forall i, k$ **do** $u_{ik} \leftarrow 1 / \sum_{j=1}^c \left(\frac{\|x_i - v_k\|}{\|x_i - v_j\|} \right)^{\frac{2}{m-1}}$
6: **end for**
7: **until** $\|\mathbf{U}^{new} - \mathbf{U}^{old}\| < \epsilon$ **return** \mathbf{V} and \mathbf{U}

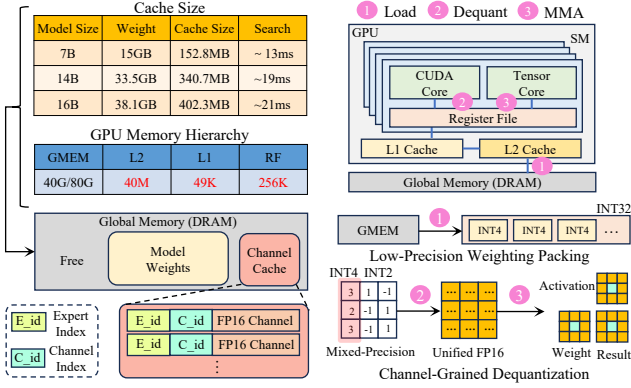


Fig. 7: Implementation Details of *DynaMo*

retrieval of target channels in the quantized MoE. We then replace the quantized channels with the cached FP16 channels whenever the indexes match. The replaced FP16 channels provide the basis for quantization switching, overcoming the limitation of upward precision adjustment.

Channel Weights Requantization for New Datasets. After channel weight replacement, we calculate the expert significance for the new dataset ($\{\mathcal{S}_{\text{exp}}^j\}_{j=1}^N \parallel \mathcal{D}_{\text{new}}$) based on Eq. 1. Then, we apply Alg. 1 to cluster the expert significance for the new dataset ($\{\mathcal{S}_{\text{exp}}^j\}_{j=1}^N \parallel \mathcal{D}_{\text{new}}$) and compare this result with the clustering results in Fig. 5. Based on the differences between the two clustering results, we can determine the specific precision to which the FP16 channels within each expert should be requantized. For instance, under the joint distribution, the 28th expert belongs to Cluster-1 (INT8); however, with the new dataset, this 28th expert is reassigned into Cluster-3 (INT4). In this case, the FP16 channels of the 28th expert should be requantized to the INT4 format. Following this approach, we requantize all FP16 channels to the appropriate precision. It is worth noting that the precision of these channels after requantization may be either higher or lower than before, thereby allowing fine-tuning of each expert’s performance, with a negligible overhead.

V. QUANTIZATION FRAMEWORK IMPLEMENTATION

Due to the expert-level mixed precision and channel-level dynamic switching mechanism, *DynaMo* encounters challenges in hardware compatibility and computational efficiency. This section will detail how *DynaMo* addresses these challenges.

A. Quantization Switch based on Channel Weight Caching

Storage Consideration for the Channel Cache. Fig. 7 presents the cache size and search time required for common MoE models (7B~16B). Among GPU memory components, only global memory (GMEM) accommodates cache, so we allocate a dedicated GMEM space for the channel cache. As we select merely 1% of channels, the cache size is negligible. For example, in a 7B MoE, the cache requires only 152.8MB, equivalent to 1.02% of the MoE’s weights storage. Notably, the processes of constructing the channel cache and calculating expert significance come before quantization adjustment, ensuring no additional overhead.

Data Format in the Channel Cache. We use the expert indexes and channel indexes as data headers and store them in the channel cache. This method ensures that, during replacement, the corresponding FP16 Cache can be quickly retrieved based on the expert indexes and channel indexes, thereby significantly reducing the overhead of online dynamic adjustment.

B. Kernel Optimization under Mix-Precision Quantization

Quantized Weight Packing Load. We use packing to enable efficient loading of low-precision weights from GMEM to the GPU. Specifically, we pack multiple low-precision data formats into an INT32, allowing simultaneous loading of multiple weights via original hardware instructions. Post-loading, we unpack the data within the GPU. This method indirectly improves hardware bandwidth utilization and constitutes the primary performance driver for inference acceleration.

Channel-Grained Dequantization. Mixed-precision quantization may lead to computational challenges, as data in different formats cannot be directly processed. To address this, we uniformly dequantize the loaded low-precision weights into FP16 using CUDA cores. Both weights and activations in FP16 can then execute MMA on tensor cores, resolving the computational issue. Furthermore, we fuse the dequantization and MMA kernels to enhance overall computational efficiency.

VI. EXPERIMENTS

A. Experiments Setup

To evaluate *DynaMo*, we select GPTQ [10] and MoEPTQ [17] as baselines. We test *DynaMo* on various MoEs [19], [25]–[27]. We test *DynaMo* under various tasks and input data distributions, including WikiText2 [23], C4 [24], ARC [28], RTE [29], PIQA [30], COPA [31], and CB [32]. All experiments are performed on NVIDIA A100 GPUs.

B. Evaluations on Language Modeling & Zero-Shot Tasks

We evaluate *DynaMo* on multiple language modeling tasks. Results in Tab. I shows that at an average ~3-bit precision, *DynaMo* has a 2.78~4.54 decrease in perplexity (PPL) compared to baseline. Next, we test *DynaMo* on multiple zero-shot inference tasks. As shown in Tab. II, *DynaMo* improves accuracy

TABLE I: Evaluations on Language Modeling Tasks.

Model	Method	#Bits	Wiki	C4	Avg. (↓)
OLMoE	w/o Quant	16	7.41	11.42	9.42
	GPTQ	3	11.65	18.86	15.26
	MoEPTQ	3.26	15.44	26.04	20.74
1B / 7B	<i>DynaMo</i>	2.95	9.64	15.31	12.48 ↓ 2.78
MoE-Girl	w/o Quant	16	8.43	13.13	10.78
	GPTQ	3	12.77	21.38	17.08
	MoEPTQ	3.26	16.88	29.40	23.14
1B / 7B	<i>DynaMo</i>	2.89	10.47	17.87	14.17 ↓ 2.91
Qwen1.5	w/o Quant	16	7.02	10.03	8.53
	GPTQ	3	10.99	20.84	15.92
	MoEPTQ	3.35	9.93	18.49	14.21
3B / 14B	<i>DynaMo</i>	3.05	8.51	14.24	11.38 ↓ 4.54
DS-MoE	w/o Quant	16	7.36	9.22	8.29
	GPTQ	3	10.47	15.19	12.83
	MoEPTQ	3.31	8.49	15.61	12.05
3B / 16B	<i>DynaMo</i>	2.98	7.94	11.49	9.72 ↓ 3.11

TABLE II: Evaluations on Zero-Shot Inference Tasks

Model	Method	#Bits	ARC-challenge	ARC-easy	RTE	PIQA	COPA	CB	Avg. (↑)
OLMoE	w/o Quant	16	29.69%	48.48%	54.51%	61.86%	71.00%	41.07%	51.10%
	GPTQ	3	25.34%	41.12%	51.99%	58.81%	62.00%	39.29%	46.43%
	MoEPTQ	3.26	24.83%	38.38%	50.54%	56.86%	65.00%	42.86%	42.27%
1B / 7B	<i>DynaMo</i>	2.97	25.85%	43.52%	54.15%	58.87%	65.00%	46.43%	48.94%
									↑ 2.54%
MoE-Girl	w/o Quant	16	31.31%	50.84%	55.95%	62.62%	66.00%	41.07%	51.30%
	GPTQ	3	25.17%	38.80%	55.59%	60.33%	62.00%	39.28%	46.86%
	MoEPTQ	3.26	24.23%	37.04%	53.06%	57.88%	59.00%	41.07%	45.38%
1B / 7B	<i>DynaMo</i>	2.88	26.02%	44.87%	53.43%	61.32%	62.00%	44.64%	48.71%
									↑ 1.85%
Qwen1.5-MoE	w/o Quant	16	33.11%	51.30%	71.84%	72.47%	81.00%	25.01%	55.79%
	GPTQ	3	26.54%	39.44%	54.51%	63.87%	72.00%	24.76%	46.85%
	MoEPTQ	3.36	24.23%	37.04%	53.07%	63.22%	67.00%	24.37%	44.82%
3B / 14B	<i>DynaMo</i>	3.04	27.13%	43.69%	55.23%	68.72%	75.00%	33.93%	50.62%
									↑ 3.77%
DeepSeek-MoE	w/o Quant	16	40.61%	71.55%	54.51%	76.22%	82.00%	41.07%	60.99%
	GPTQ	3	33.62%	62.04%	52.34%	73.94%	78.00%	44.64%	57.43%
	MoEPTQ	3.36	31.99%	60.06%	53.43%	69.42%	77.00%	41.07%	55.50%
3B / 16B	<i>DynaMo</i>	2.99	34.71%	64.19%	53.92%	75.34%	81.00%	47.87%	59.51%
									↑ 2.08%

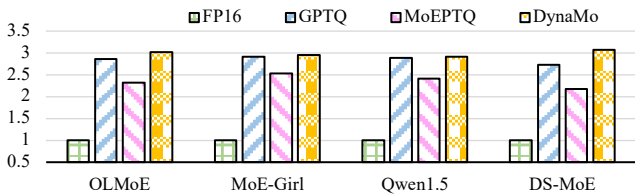


Fig. 8: Speedup of *DynaMo* on Various MoEs

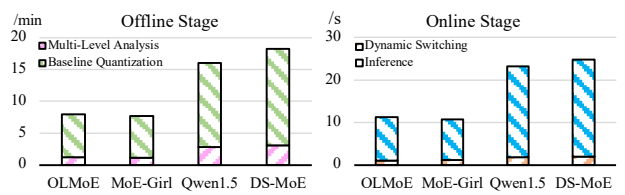


Fig. 9: Offline/Online Stage Overhead of *DynaMo*

by 1.85%~3.77% over baselines. These tests demonstrate that *DynaMo* can adapt to various datasets while achieving lower quantization loss at the same compression ratio.

C. Results of *DynaMo*'s Inference Speed

The inference speed of *DynaMo* are shown in Fig. 8. Compared to FP16 model, *DynaMo* achieve 2.91~3.08× speedup. Compared with baselines, *DynaMo* still shows slight speed improvements. Existing research shows quantization's primary performance boost stems from loading low-precision data from GMEM to the GPU. Though mixed precision adds extra computational overhead, we maximize bandwidth utilization by packing low-precision data during loading, enabling inference speed on par with SOTA methods. Note that the key advantage of *DynaMo* lies in its flexibility and adjustability when addressing variations across datasets. Its core focus is on guaranteeing accuracy across diverse datasets, rather than chasing speed.

D. Ablation Study

We conducted an ablation study on *DynaMo*, using Wiki-Text2 for language modeling and CB for zero-shot tasks.

TABLE III: Ablation Study

Model	Task	GPTQ	Only BQ	+ DQS
OLMoE	Wiki	11.65	10.78 ↓ 0.87	9.64 ↓↓ 1.23
	CB	39.29%	42.79% ↑ 3.50%	46.43% ↑↑ 3.64%
MoE-Girl	Wiki	12.77	11.23 ↓ 1.54	10.47 ↓↓ 0.76
	CB	39.28%	42.66% ↑ 3.38%	44.64% ↑↑ 1.98%
Qwen1.5	Wiki	10.99	9.82 ↓ 1.17	8.51 ↓↓ 1.31
	CB	24.76%	29.69% ↑ 4.93%	33.93% ↑↑ 4.24%
DS-MoE	Wiki	10.47	8.52 ↓ 1.95	7.94 ↓↓ 0.58
	CB	44.64%	45.91% ↑ 1.27%	47.87% ↑↑ 1.96%

First, we performed only expert-level baseline quantization (BQ) on the C4 and RTE datasets, then compared *DynaMo* and GPTQ. Next, we added channel-level dynamic switching (DQS) based on WikiText2 and CB, and retested. All results are shown in Tab. III. With Only BQ, PPL decreased by 0.87~1.95 and accuracy improved by 1.27%~4.93%, confirming our multi-stage analysis accurately captures MoE's inherent dynamics, identifies optimal expert quantization precision, and mitigates quantization-induced accuracy loss. Adding +DQS further reduced PPL by 0.58~1.31 and boosted accuracy by 1.96%~4.24%, showing dynamic switching helps quantized MoE adapt to new datasets for better performance.

E. Overhead Discussion

We evaluate *DynaMo*'s overhead. Given that multi-level analysis and baseline quantization are offline processes, while dynamic quantization switching and model inference are online, we test them separately. As shown in Fig. 9, multi-stage analysis accounts for 14.90%~17.33% of offline phase overhead, and dynamic quantization switching only takes 7.73%~10.68% of online overhead. Note that inference time here refers to the duration for MoE to generate 1024 tokens; additionally, dynamic quantization switching is only needed when the dataset changes. Overall, *DynaMo*'s cost is negligible.

VII. CONCLUSION

This paper proposes *DynaMo*, a novel MoE quantization framework. First, it enables expert-level mixed-precision baseline quantization compatible with multiple existing datasets; then, it introduces channel-level dynamic switching to adapt quantized MoEs to new datasets. Experiments show that *DynaMo* outperforms SOTA works in quantization performance.

REFERENCES

- [1] W. Cai, J. Jiang, F. Wang, J. Tang, S. Kim, and J. Huang, "A survey on mixture of experts," 2024. [Online]. Available: <https://arxiv.org/abs/2407.06204>
- [2] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: scaling to trillion parameter models with simple and efficient sparsity," *J. Mach. Learn. Res.*, vol. 23, no. 1, Jan. 2022.
- [3] W. Gan, Z. Ning, Z. Qi, and P. S. Yu, "Mixture of experts (moe): A big data perspective," *arXiv preprint arXiv:2501.16352*, 2025.
- [4] P. Li, R. Li, Q. Da, A. X. Zeng, and L. Zhang, "Improving multi-scenario learning to rank in e-commerce by exploiting task relationships in the label space," in *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management*, 2020.
- [5] X. Wang, J. Cao, Z. Fu, K. Gai, and G. Zhou, "Home: Hierarchy of multi-gate experts for multi-task learning at kuaishou," 2024. [Online]. Available: <https://arxiv.org/abs/2408.05430>
- [6] H. Mei, D. Cai, A. Zhou, S. Wang, and M. Xu, "Fedmoe: Personalized federated learning via heterogeneous mixture of experts," 2024. [Online]. Available: <https://arxiv.org/abs/2408.11304>
- [7] M. A. Aghdam, H. Jin, and Y. Wu, "Da-moe: Towards dynamic expert allocation for mixture-of-experts models," 2024. [Online]. Available: <https://arxiv.org/abs/2409.06669>
- [8] S. Zhong, Y. Sun, L. Liang, R. Wang, R. Huang, and M. Li, "Hybrimoe: Hybrid cpu-gpu scheduling and cache management for efficient moe inference," 2025. [Online]. Available: <https://arxiv.org/abs/2504.05897>
- [9] R. Gong, Y. Ding, Z. Wang, C. Lv, X. Zheng, J. Du, H. Qin, J. Guo, M. Magno, and X. Liu, "A survey of low-bit large language models: Basics, systems, and algorithms," 2024. [Online]. Available: <https://arxiv.org/abs/2409.16694>
- [10] E. Frantar, S. Ashkboos, T. Hoefer, and D. Alistarh, "Gptq: Accurate post-training quantization for generative pre-trained transformers," 2023. [Online]. Available: <https://arxiv.org/abs/2210.17323>
- [11] K. Behdin, A. Acharya, A. Gupta, Q. Song, S. Zhu, S. Keerthi, and R. Mazumder, "Quantease: Optimization-based quantization for language models," 2023. [Online]. Available: <https://arxiv.org/abs/2309.01885>
- [12] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, "Smoothquant: Accurate and efficient post-training quantization for large language models," 2024. [Online]. Available: <https://arxiv.org/abs/2211.10438>
- [13] 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 llm compression and acceleration," 2024. [Online]. Available: <https://arxiv.org/abs/2306.00978>
- [14] Z. Yuan, L. Niu, J. Liu, W. Liu, X. Wang, Y. Shang, G. Sun, Q. Wu, J. Wu, and B. Wu, "Rptq: Reorder-based post-training quantization for large language models," 2023. [Online]. Available: <https://arxiv.org/abs/2304.01089>
- [15] Y. Zhao, C.-Y. Lin, K. Zhu, Z. Ye, L. Chen, S. Zheng, L. Ceze, A. Krishnamurthy, T. Chen, and B. Kasicki, "Atom: Low-bit quantization for efficient and accurate llm serving," 2024. [Online]. Available: <https://arxiv.org/abs/2310.19102>
- [16] Y. J. Kim, R. Fahim, and H. H. Awadalla, "Mixture of quantized experts (moqe): Complementary effect of low-bit quantization and robustness," 2023. [Online]. Available: <https://arxiv.org/abs/2310.02410>
- [17] P. Li, X. Jin, Y. Cheng, and T. Chen, "Examining post-training quantization for mixture-of-experts: A benchmark," 2024. [Online]. Available: <https://arxiv.org/abs/2406.08155>
- [18] Y. Shen, Z. Guo, T. Cai, and Z. Qin, "Jetmoe: Reaching llama2 performance with 0.1m dollars," 2024. [Online]. Available: <https://arxiv.org/abs/2404.07413>
- [19] N. Muennighoff, L. Soldaini, D. Groeneveld, K. Lo, J. Morrison, S. Min, W. Shi, P. Walsh, O. Tafjord, N. Lambert, Y. Gu, S. Arora, A. Bhagia, D. Schwenk, D. Wadden, A. Wettig, B. Hui, T. Dettmers, D. Kiela, A. Farhadi, N. A. Smith, P. W. Koh, A. Singh, and H. Hajishirzi, "Olmoe: Open mixture-of-experts language models," 2024. [Online]. Available: <https://arxiv.org/abs/2409.02060>
- [20] X. Pan, W. Lin, L. Zhang, S. Shi, Z. Tang, R. Wang, B. Li, and X. Chu, "Fsmoe: A flexible and scalable training system for sparse mixture-of-experts models," *arXiv preprint arXiv:2501.10714*, 2025.
- [21] H. Hazimeh, Z. Zhao, A. Chowdhery, M. Sathiamoorthy, Y. Chen, R. Mazumder, L. Hong, and E. Chi, "Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 335–29 347, 2021.
- [22] S. Zhong, L. Liang, Y. Wang, R. Wang, R. Huang, and M. Li, "Adapmoe: Adaptive sensitivity-based expert gating and management for efficient moe inference," in *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, 2024, pp. 1–9.
- [23] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: <https://openreview.net/forum?id=Byj72udxe>
- [24] 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," *J. Mach. Learn. Res.*, vol. 21, pp. 140:1–140:67, 2020. [Online]. Available: <https://jmlr.org/papers/v21/20-074.html>
- [25] Allura-org, "Moe-girl," 2024. [Online]. Available: <https://huggingface.co/allura-org/MoE-Girl-1BA-7BT>
- [26] Q. Team, "Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters," February 2024. [Online]. Available: <https://qwenlm.github.io/blog/qwen-moe/>
- [27] D. Dai, C. Deng, C. Zhao, R. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu *et al.*, "Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models," *arXiv preprint arXiv:2401.06066*, 2024.
- [28] M. Boratko, H. Padigela, D. Mikkilineni, P. Yuvraj, R. Das, A. McCallum, M. Chang, A. Fokoue-Nkoutche, P. Kapanipathi, and N. Mattei, "A systematic classification of knowledge, reasoning, and context within the arc dataset," 2018.
- [29] K. Wang, Z. Liu, Y. Lin, J. Lin, and S. Han, "Haq: Hardware-aware automated quantization with mixed precision," 2019.
- [30] S. Tata and J. M. Patel, "Piqa: An algebra for querying protein data sets," in *Proceedings of the 15th International Conference on Scientific and Statistical Database Management (SSDBM 2003), 9-11 July 2003, Cambridge, MA, USA*. IEEE Computer Society, 2003, pp. 141–150. [Online]. Available: <https://doi.org/10.1109/SSDM.2003.1214975>
- [31] M. Roemmele, C. A. Bejan, and A. S. Gordon, "Choice of plausible alternatives: An evaluation of commonsense causal reasoning," in *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06, Stanford, California, USA, March 21-23, 2011*. AAAI, 2011. [Online]. Available: <http://www.aaai.org/ocs/index.php/SSS/SSS11/paper/view/2418>
- [32] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "Superglue: A stickier benchmark for general-purpose language understanding systems," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: <https://proceedings.neurips.cc/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf>