Sub-MoE: Efficient Mixture-of-Expert LLMs Compression via Subspace Expert Merging

Lujun Li[1*], **Qiyuan Zhu**[1*], **Jiacheng Wang**[2], **Xiaoyu Qin**[3], **Wei Li**[4], **Hao Gu**[1], **Sirui Han**[1†], **Yike Guo** [1 †]

[1]The Hong Kong University of Science and Technology
[2]Xi'an Jiaotong University
[3]Tsinghua University
[4]University of Birmingham
{lliee,qzhuat,hguam}@connect.ust.hk, {siruihan,yikeguo}@ust.hk, jiacheng@stu.xjtu.edu.cn, xiao.y.qin@gmail.com, wxl885@student.ham.ac.uk

Abstract

Mixture of Experts (MoE) LLMs face significant obstacles due to their massive parameter scale, which imposes memory, storage, and deployment challenges. Although recent expert merging methods aim to achieve greater efficiency by consolidating several experts, they are fundamentally hindered by parameter conflicts arising from expert specialization. In this paper, we present Sub-MoE, a novel MoE compression framework via Subspace Expert Merging. Our key insight is to perform joint Singular Value Decomposition (SVD) on concatenated expert weights, reducing conflicting parameters by extracting shared U-matrices while enabling effective merging of the expert-specific V components. Specifically, Sub-MoE consists of two innovative stages: (1) Adaptive Expert Clustering, which groups functionally coherent experts via K-means clustering based on cosine similarity of expert outputs; and (2) Subspace Expert Merging, which first performs Experts Union Decomposition to derive the shared U-matrix across experts in the same group, then applies frequency-based merging for individual V-matrices, and completes expert reconstruction using the merged V-matrix. In this way, we align and fuse experts in a shared subspace. Additionally, the framework can be extended with intra-expert compression for further inference optimization. Extensive experiments on Mixtral, DeepSeek, and Qwen-1.5/3 MoE LLMs demonstrate that our Sub-MoE significantly outperforms existing expert pruning and merging methods. Notably, our Sub-MoE maintains 96%/86% of original performance with 25%/50% expert reduction on Mixtral-8×7B in zero-shot benchmarks.

1 Introduction

The Mixture of Experts (MoE) architecture has emerged as a pivotal advancement in Large Language Models (LLMs) (Nguyen et al. 2024), demonstrated by recent models like DeepSeek-R1 (DeepSeek-AI et al. 2024) and Qwen3-MoE (Yang et al. 2024a). At its core, MoE consists of expert networks and a gating mechanism that dynamically routes each input to the most relevant experts. MoE sparsely activates only a small subset of experts, significantly reducing computational costs while scaling model size. However, MoE LLMs also introduce challenges from their large parameter count, including substantial memory/storage requirements and inference latency that complicate deployment on resource-constrained devices (Song et al. 2023; Liu, Wang, and Wu 2025). Additionally, distributed implementations face communication (Jiang et al. 2024b) bottlenecks when synchronizing experts across multiple nodes, impacting real-time performance (Shen et al. 2022).

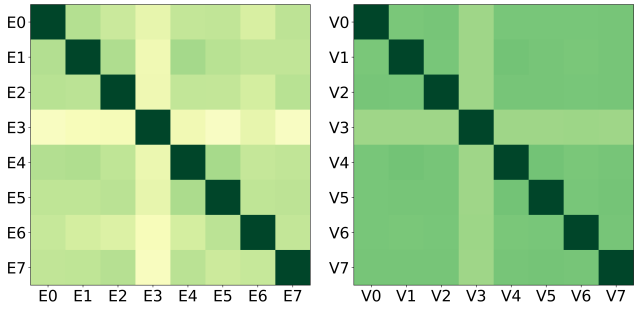

Figure 1: Cosine similarity of output of original expert (left) and subspace aligned matrices (right) on Mixtral-8×7B. Colour bar ranging from yellow to green denotes a numerical transition from 0 to 1.

To overcome these issues, researchers are developing fundamental yet important expert reduction approaches that can be broadly categorized into two primary approaches: expert pruning and expert merging. Expert pruning methods remove underperforming experts through regularization (*e.g.*, SEER-MoE (Muzio, Sun, and He 2024)) or search-based techniques (*e.g.*, NAEE (Lu et al. 2024), MoE-I^2 (Yang et al. 2024b)). While these approaches effectively reduce parameter counts, they fundamentally discard portions of the model's learned knowledge, resulting in performance degradation that necessitates resource-intensive fine-tuning to re-

[*]These authors contributed equally.
[†]Corresponding author

cover. Expert merging techniques (*e.g.*, MC-SMoE (Li et al. 2023b), HC-SMoE (Chen et al. 2024), and EEP (Liu et al. 2024)) propose a promising alternative by preserving knowledge through the consolidation of several experts. However, current merging approaches encounter a critical limitation that undermines their effectiveness: the parameter conflict problem. This fundamental challenge arises from the core design principle of MoE architectures, where routing mechanisms deliberately create specialized experts with divergent parameter spaces by training them on distinct input distributions. The recent study (Gu et al. 2025b) shows that Mixtral-8×7B demonstrates this divergence, showing inter-expert similarities typically ranging between 0.1∼0.3. When conventional merging operations are applied to such dissimilar experts, catastrophic parameter conflicts emerge that compromise the specialized capabilities of the original experts and significantly degrade overall model performance. Existing merging approaches employ simplistic aggregation functions that cannot effectively reconcile these divergent parameter spaces and often require computationally expensive post-merging operations (*e.g.*, D^2-MoE (Gu et al. 2025b)), undermining the efficiency gains. This motivates our core research question:

(RQ) How can we reduce parameter conflicts among diverse experts and enhance the effectiveness of expert merging?

To answer the question, we present Sub-MoE, a novel expert merging framework rooted in subspace-based decomposition and alignment. Our approach leverages Singular Value Decomposition (SVD) to transform the concatenated weight matrices of multiple experts into a shared low-dimensional subspace, represented by a common orthogonal basis U, singular values Σ, and individual projections V. By performing the merging operation solely on the V component, while preserving alignment to the shared U, we exploit intrinsic correlations among experts, thereby minimizing conflicting parameters and retaining specialized knowledge. As illustrated in Figure 2, our Sub-MoE framework consists of two synergistic stages: Adaptive Expert Clustering and Subspace Expert Merging. In the first stage, we perform adaptive expert grouping via K-means clustering based on output similarities of experts, ensuring that merging is performed on functionally coherent groups. In addition, we jointly cluster multi-layer experts under a target overall compression ratio and adaptively determine the layer-wise grouping numbers. In the second stage, we concatenate expert weights from the same group and perform co-decomposition to obtain the shared U-matrix across experts and expert-specific V-matrices. Figure 1 demonstrates that this subspace-sharing process can align the output of the various experts. For the remaining unmerged components, we further introduce the frequency-based merging strategy that weights expert contributions according to their activation patterns. This approach gives greater influence to frequently activated experts while still preserving capabilities from all experts. Finally, the merged weight matrix is reconstructed as $U\Sigma[V_{\text{merged}}]^T$ [1]. Additionally, we extend Sub-MoE to Sub-MoE† with MoE-specific activation-aware truncated SVD for intra-expert compression for greater parameter efficiency. We incorporate input activation statistics by weighting expert parameters with the whitening matrix of hidden activations, further stabilizing performance at high compression levels.

We conduct comprehensive experiments on Mixtral 8x7B (Jiang et al. 2024a), Qwen3-30B-A3B (Yang et al. 2024a) Qwen1.5-MoE-A2.7B (Team 2024) and DeepSeekMoE-16B-Base (Dai et al. 2024). Our proposed Sub-MoE method consistently outperforms existing expert reduction techniques. With Mixtral-8×7B, Sub-MoE maintains 94% and 87% of accuracy using only 75% or 50% of experts, surpassing HC-SMoE(Chen et al. 2024) by 13.7%. For Qwen3 MoE, Sub-MoE maintains 83% of accuracy with half the experts, while HC-SMoE drops to 55%. Similarly, on DeepSeek-MoE-16B, our method preserves 86% of performance with half the experts, outperforming HC-SMoE by 6.5%. These results affirm the effectiveness and generalizability of our approach across diverse MoE architectures and downstream tasks, establishing Sub-MoE as a principled and scalable solution for expert merging in MoE LLMs.

2 Related Work

MoE Compression. To improve the efficiency of MoE LLMs, researchers have developed numerous system-level optimizations (*e.g.*, expert parallel (Cai et al. 2024) and offloading (Xue et al. 2024)) and model-level techniques (MoE-specific quantization (Huang et al. 2025) and compression (Sarkar et al. 2024)). Among them, expert reduction methods primarily focus on removing redundant experts to achieve optimal efficiency-performance tradeoffs. For optimization-based expert pruning, TSEP (Chen et al. 2022) and SEER-MoE (Muzio, Sun, and He 2024) remove non-professional experts through regularization-based fine-tuning. In search-based expert pruning approaches, NAEE (Lu et al. 2024) trims unimportant experts by minimizing pruning error, while MoE-I^2 (Yang et al. 2024b) employs genetic search strategies. **In sharp contrast to these pruning approaches, our Sub-MoE explores the merging paradigm that requires neither searching nor fine-tuning.** Other methods use weight or hybrid compression for MoE. MoE-Pruner (Xie et al. 2024) prunes weights based on activations and router logits, STUN (Lee et al. 2024) combines structured and unstructured pruning, and D^2-MoE (Gu et al. 2025b) introduces delta compensation. MoE-Compression (He et al. 2024) provides compressor evaluations. **Different from these, we highlight that Sub-MoE mainly addresses expert merging rather than weight compression.**

Expert Merging methods (Li et al. 2022; Zhao et al. 2024) fuse multiple experts into a single one through weighted summation or averaging. For instance, MC-SMoE (Li et al. 2023b) merges experts with similar routing policies, HC-SMoE (Chen et al. 2024) utilizes hierarchical clustering to merge experts in a task-agnostic manner, and EEP (Liu et al. 2024) optimizes fusion matrices through evolutionary

[1]The singular values Σ are multiplied in the shared U matrix during the joint SVD process in implementation.

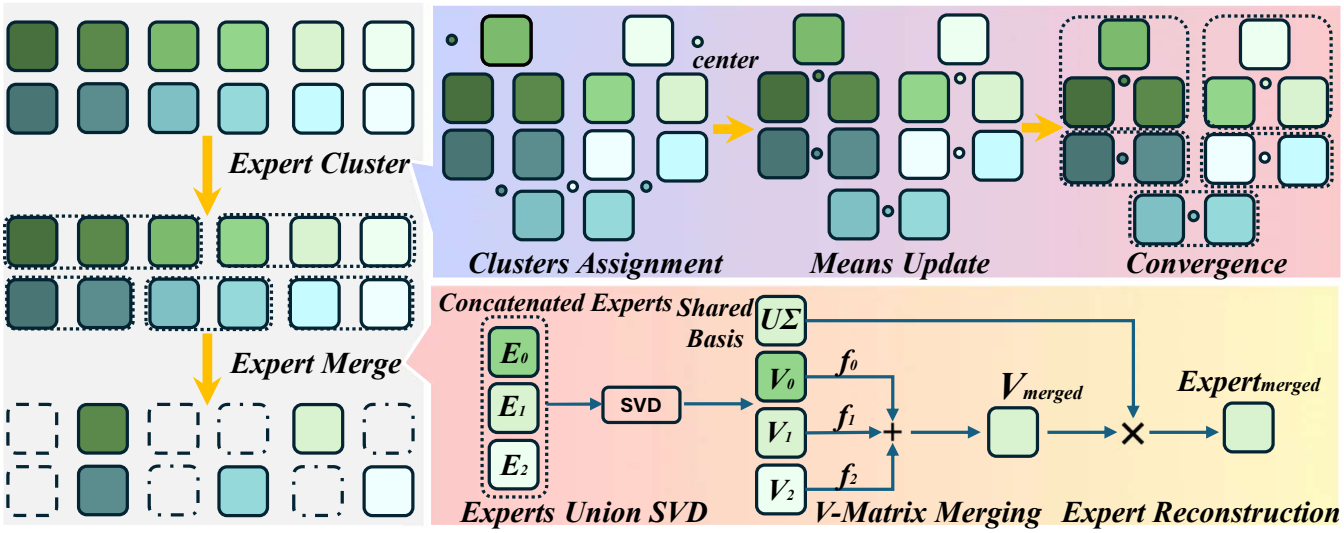

Figure 2: Overview of Sub-MoE framework, which consists of two main stages: (1) Adaptive Expert Clustering, which groups similar experts via K-means clustering with steps: Clusters Assignment, Means Update, and Convergence; and (2) Subspace Expert Merging, which aligns and combines experts via Experts Union SVD, V-Matrix Merging. and Expert Reconstruction.

search algorithms. However, these methods typically employ original weight merging techniques (Wortsman et al. 2022; Matena and Raffel 2022), which achieve success primarily when handling models with high similarity, such as fine-tuned variants of the same base model (Izmailov et al. 2018). When applied to MoE models with low-similarity experts, these methods generally fail due to significant parameter conflicts during the merging process. **Our Sub-MoE distinctly differs from previous expert mergers** by addressing this low-similarity issue through expert decomposition into subspaces and ensuring matrices alignment. Thanks to this subspace alignment approach, we can effectively fuse different MoE LLMs without requiring additional training, searching, or complex weight operations.

3 Methodology

Our Sub-MoE framework consists of two synergistic stages: (1) **Adaptive Expert Clustering** stage that clusters similar experts, (2) **Subspace Expert Merging** stage that includes union decomposition, frequency-based V-Matrix fusion and reconstruction. The overall process is illustrated in Figure 2.

3.1 Recap of MoE Architecture

The fundamental principle of MoE models is to dynamically route input data to specialized expert networks. Consider an input token $x \in \mathbb{R}^d$, a set of expert modules $\{E_1, E_2, ..., E_n\}$, and a router network R. The output y of an MoE layer is computed as:

$$y = \sum_{i=1}^{n} G_i(x) \cdot E_i(x), E(x) = (\sigma(x \cdot W_{gate}) \odot (x \cdot W_{up})) \cdot W_{down} \tag{1}$$

where $G_i(x)$ represents the routing score for expert i, and $E_i(x)$ denotes its output. Each expert typically implements a feed-forward layer with weight matrices $\{W_{\text{up}}, W_{\text{gate}}, W_{\text{down}}\}$, and σ activation (*e.g.*, SiLU function). The router R employs a top-k strategy with softmax normalization, activating only the most relevant experts for each input token and thereby enhancing computational efficiency.

3.2 Adaptive Expert Clustering

A critical challenge in compressing MoE models is identifying which experts can be effectively merged with minimal information loss. Rather than relying on architectural heuristics or arbitrary grouping strategies, we propose a data-driven approach that captures the functional similarity between experts. Our key insight is that experts processing similar input patterns in comparable ways are more amenable to merging. To implement this intuition, we first collect a representative set of input tokens $\mathcal{X} = \{x_1, x_2, ..., x_m\}$ from the target domain. For each expert E_i, we compute output vectors across this input set, yielding output collections $\mathcal{Y}_i = \{E_i(x_1), E_i(x_2), ..., E_i(x_m)\}$ that characterize the expert's functional behavior. We then quantify the functional similarity between experts using the average cosine similarity of their outputs:

$$\text{Sim}(E_i, E_j) = \frac{1}{m} \sum_{l=1}^{m} \frac{E_i(x_l) \cdot E_j(x_l)}{||E_i(x_l)|| \cdot ||E_j(x_l)||}, \tag{2}$$

This similarity metric captures how consistently two experts respond to the same inputs, regardless of their internal parameter representations. Experts with high similarity scores are likely to serve overlapping functions within the model and thus become strong candidates for merging.

Based on this similarity measure, we employ K-means clustering to organize the experts into k coherent groups. This process consists of four key steps: **(1). Means Initialization:** Initial cluster centroids $C = \{C_1, C_2, ..., C_k\}$ are established through an advanced seeding method (*i.e.*,

k-means++ (Ikotun et al. 2023)) to ensure diverse starting points across the expert functional space. **(2). Clusters Assignment:** Each expert E_j is assigned to the nearest cluster centroid based on the similarity metric in Equation 2, forming expert groups Q_i that share functional characteristics. **(3). Means Update:** Cluster centroids are recalculated as the mean of all experts assigned to that cluster: $C_i = \frac{1}{|Q_i|} \sum_{E_j \in Q_i} \mathcal{Y}_j$. **(4). Convergence:** Steps 2 and 3 are repeated until cluster assignments stabilize or maximum iterations are reached, minimizing the objective function:

$$J = \sum_{i=1}^{k} \sum_{E_j \in Q_i} ||\mathcal{Y}_j - C_i||^2, \tag{3}$$

where Q_i represents the set of experts assigned to cluster i. This data-driven approach discovers inherent functional relationships between experts that might not be apparent from architecture alone.

Multi-layer Adaptive Allocation: Unlike traditional manners that impose uniform reduction across all MoE layers, we introduce a multi-layer adaptive allocation that optimizes the numbers of groups on a per-layer basis. We recognize that different layers within a model exhibit varying degrees of functional redundancy and specialization. By jointly clustering experts on multiple MoE layers while maintaining a target overall compression ratio, our automated clustering process dynamically adjusts clustering centers and determines the optimal number of clusters for each layer without manual intervention. Layers with higher expert similarity naturally form fewer, more cohesive clusters, while those with more diverse patterns maintain more clusters to preserve their specialized capabilities.

3.3 Subspace Expert Merging

Challenge in merging expert networks lies in their different parametric representations. Even when experts serve similar functions, their internal parameters often operate in distinct representation spaces, making direct merging problematic and leading to performance degradation. Given n expert weight matrices $W^{(1)}, W^{(2)}, ..., W^{(n)} \in \mathbb{R}^{O \times I}$, conventional merging methods apply operations directly:

$$W_{\text{merged}} = \sum_{i=1}^{n} \alpha_i W^{(i)}, \tag{4}$$

where α_i are weight coefficients. This approach often leads to parameter conflicts because each $W^{(i)}$ operates in its own representation space.

Subspace Alignment via Experts Union Decomposition. We address this challenge by transforming experts into a common subspace before merging. For each expert group identified in the clustering step, we separately concatenate their W_{gate}, W_{up} and W_{down} weight matrices and apply SVD:

$$\text{SVD}\left([W'^{(1)}; \ldots; W'^{(n)}]\right) = U' \Sigma' [V'^{(1)}; \ldots; V'^{(n)}]^T \tag{5}$$

where $U \in \mathbb{R}^{O \times r}$ contains left singular vectors, which form an orthonormal basis for the input space, $\Sigma \in \mathbb{R}^{r \times r}$ is a diagonal matrix of singular values, and $V \in \mathbb{R}^{r \times nI}$ contains right singular vectors, which can be partitioned into n blocks, each corresponding to an expert.

Frequency-based V-Matrix Merging. Our method introduces a simple yet effective merging approach that respects the usage patterns of experts in real-world scenarios. We observe that not all experts contribute equally to model outputs–some experts specialize in handling common patterns while others focus on rare cases. Incorporating this frequency information helps preserve the model's capabilities across diverse inputs. For each expert i, we calculate its sampling frequency based on actual router activations:

$$f(V_i) = \frac{\sum_{x \in \mathcal{X}} \mathbb{I}[i \in \text{TopK}(G(x), k)]}{|\mathcal{X}|}, \tag{6}$$

where $\mathcal{X}$ represents the set of input tokens, $\mathbb{I}[\cdot]$ is the indicator function that equals 1 when expert i is among the top-k experts selected by the routing mechanism for input x, and 0 otherwise. This frequency metric captures how often each expert is activated across a representative dataset. We then compute the merged V matrix in each group as a frequency-weighted average after applying TIES-based(Yadav et al. 2023) sparsification to V_i to reduce parameter conflicts:

$$V_{\text{merged}} = \frac{\sum_{i \in Q} f(V_i) \cdot V_i}{\sum_{i \in Q} f(V_i)}, \tag{7}$$

This frequency-based merging effectively integrates expert information according to their practical utilization patterns, giving greater weight to frequently activated experts while still preserving capabilities from all experts.

Expert Reconstruction. The final merged expert weights are constructed as:

$$W_{\text{merged}} = U \Sigma [V_{\text{merged}}]^T, \tag{8}$$

By construction, all experts within a cluster are merged into a single set of parameters W_{merged}, which is reconstructed using the shared orthogonal basis U, singular values Σ, and the frequency-weighted merged right singular vectors V_{merged}. This process effectively aligns the original experts to a common subspace and compresses them into one representative expert. With this three-stage process, Sub-MoE achieves effective expert reduction by operating in a shared subspace, minimizing parameter conflicts, and preserving the essential characteristics of each expert.

3.4 Sub-MoE† for Intra-Expert Compression

Sub-MoE reduces expert counts without changes to intra-expert sizes. To further improve the compression ratio for resource-constrained scenarios, we present extended Sub-MoE† to reduce the size of U, V by truncating before reconstruction. Beyond previous dense LLM SVD techniques (Yuan et al. 2023; Wang et al. 2024), our Sub-MoE† employs MoE-specific activation-aware truncating SVD. For expert weight matrix W_i, we first obtain the activation weighted matrix S_i by measuring the correlation of input activations X_i. S_i effectively preserves salience weights and reduces decomposition errors (Yuan et al. 2023; Wang et al.

Expert	Method	WikiText-2↓	PTB↓	C4↓	ARC-c	ARC-e	BoolQ	HellaS.	MMLU	OBQA	RTE	WinoG.	Average↑
						Mixtral-8×7B							
Num=8	Original	3.98	14.79	7.33	0.56	0.84	0.85	0.65	0.67	0.35	0.71	0.76	0.67
Num=6	Frequency-prune	6.22	18.00	9.94	0.48	0.78	0.78	0.57	0.47	0.32	0.55	0.75	0.59
	Output-prune	6.17	18.28	9.63	0.47	0.77	0.75	0.58	0.46	0.30	0.60	0.75	0.58
	MC-SMoE	58.11	173.51	98.86	0.29	0.60	0.59	0.43	0.25	0.20	0.53	0.60	0.44
	HC-SMoE	5.92	18.70	9.49	0.45	0.73	0.83	0.57	0.56	0.29	0.69	0.75	0.61
	Sub-MoE (Ours)	**5.16**	**18.58**	**8.54**	**0.49**	**0.80**	**0.86**	**0.62**	**0.59**	**0.32**	**0.65**	**0.75**	**0.64**
Num=4	Frequency-prune	17.45	79.43	22.40	0.22	0.39	0.60	0.36	0.24	0.14	0.53	0.53	0.38
	Output-prune	15.40	81.96	20.08	0.21	0.39	0.63	0.38	0.24	0.16	0.54	0.56	0.39
	MC-SMoE	854.05	1204.41	1408.10	0.21	0.28	0.52	0.28	0.25	0.11	0.50	0.52	0.33
	HC-SMoE	9.88	34.13	16.78	0.32	0.61	0.75	0.49	0.39	0.26	0.61	0.67	0.51
	Sub-MoE (Ours)	**6.97**	**26.88**	**10.64**	**0.45**	**0.75**	**0.84**	**0.57**	**0.48**	**0.29**	**0.57**	**0.72**	**0.58**
						Qwen1.5-MoE-A2.7B-Chat							
Num=60	Original	8.12	12.97	11.62	0.40	0.71	0.81	0.59	0.60	0.31	0.74	0.66	0.60
Num=45	HC-SMoE	12.97	17.45	16.65	0.36	0.66	0.78	0.50	0.46	0.28	0.74	0.65	0.55
	Sub-MoE (Ours)	**12.37**	17.45	**16.39**	**0.37**	**0.67**	**0.80**	**0.53**	**0.49**	**0.29**	0.71	0.64	**0.56**
Num=30	HC-SMoE	19.16	24.60	26.19	0.32	0.58	0.76	0.44	0.43	0.22	0.69	0.63	0.51
	Sub-MoE (Ours)	**17.18**	**22.06**	**22.76**	**0.35**	**0.62**	0.76	**0.47**	**0.44**	**0.23**	0.67	**0.65**	**0.52**
						Qwen3-30B-A3B							
Num=128	Original	8.64	15.40	14.47	0.53	0.80	0.89	0.60	0.78	0.35	0.83	0.71	0.69
Num=96	HC-SMoE	18.86	31.11	29.68	0.35	0.64	0.82	0.40	0.55	0.22	0.73	0.61	0.54
	Sub-MoE (Ours)	**13.59**	**23.48**	**21.38**	**0.44**	**0.70**	**0.86**	**0.47**	**0.65**	**0.25**	**0.76**	**0.66**	**0.60**
Num=64	HC-SMoE	72.33	162.99	148.41	0.23	0.44	0.63	0.29	0.30	0.13	0.50	0.50	0.38
	Sub-MoE (Ours)	**21.05**	**43.19**	**36.37**	**0.40**	**0.68**	**0.84**	**0.41**	**0.56**	**0.23**	**0.77**	**0.63**	**0.57**
						DeepSeek-MoE-16B							
Num=64	Original	6.51	9.72	10.15	0.44	0.76	0.72	0.58	0.38	0.33	0.62	0.70	0.57
Num=48	HC-SMoE	9.13	12.19	13.45	0.39	0.71	0.72	0.52	0.30	0.30	0.64	0.70	0.53
	Sub-MoE (Ours)	**8.48**	**11.29**	**12.60**	**0.40**	**0.72**	**0.73**	**0.54**	**0.32**	0.27	**0.66**	**0.70**	**0.55**
Num=32	HC-SMoE	15.34	21.07	23.30	0.31	0.60	0.69	0.43	0.24	0.20	0.57	0.64	0.46
	Sub-MoE (Ours)	**13.71**	**18.35**	**20.70**	**0.32**	**0.63**	**0.68**	**0.44**	**0.25**	**0.22**	**0.65**	**0.65**	**0.49**

Table 1: Comparisons of expert prune/merge methods in multiple MoE LLMs. We report perplexity (lower is better↓) on language modeling tasks and accuracy (higher is better↑) on reasoning tasks.

2024). Then, we re-weight each expert's weight matrix as $W_i' = W_i S_i$. Next, we concatenate the re-weighted weight matrices from all experts in the same group and apply union decomposition:

$$\text{SVD}\left([W'^{(1)}; \ldots; W'^{(n)}]\right) = U'\Sigma'[V'^{(1)}; \ldots; V'^{(n)}]^T, \quad (9)$$

For experts in cluster Q, we compute the frequency-weighted merged vector with de-whitening:

$$V_{\text{merged}} = \frac{\sum_{i \in Q} f(V_i) \cdot V'^{(i)} S_i^{-1}}{\sum_{i \in Q} f(V_i)}, \quad (10)$$

After truncating the smallest singular values in Σ' to control the compression ratio, the final merged expert weight is given by:

$$W_{\text{merged}}^{\text{trunc}} = U' \cdot \text{Trunc.}(\Sigma') \cdot V_{\text{merged}}. \quad (11)$$

This process enables fine-grained control over compression while minimizing information loss, as the activation weighted matrix S enables a direct mapping between singular values and compression loss (Wang et al. 2024). By combining these strategies Sub-MoE† provides more extreme compression for MoE LLMs.

4 Experiments

4.1 Experimental Setups

We conduct experiments on 4 MoE LLMs: Mixtral 8x7B (Jiang et al. 2024a), Qwen3-30B-A3B (Yang et al. 2024a) Qwen1.5-MoE-A2.7B (Team 2024) and DeepSeekMoE-16B-Base (Dai et al. 2024). For Mixtral 8x7B, reducing experts from 8 to 4 decreases the model size from 46.7B to 24.2B parameters and reduces computational requirements from 2989 to 1546 GFLOPs. Similarly, for Qwen1.5-MoE, reducing experts from 60 to 30 results in a 43% reduction in model size (from 14.3B to 8.1B parameters). To evaluate our method comprehensively, we use two types of metrics: (1) perplexity on standard language modeling benchmarks including WikiText-2 (Merity et al. 2016), PTB (Mikolov et al. 2010), and C4 (Raffel et al. 2023), and (2) accuracy on eight diverse reasoning and understanding tasks (Gao et al. 2023) like ARC (Clark et al. 2018), BoolQ (Clark et al. 2019), HellaSwag (Zellers et al. 2019), MMLU (Hendrycks et al. 2021), OBQA (Mihaylov et al. 2018), RTE (Bentivogli et al. 2009), and WinoG. (Sakaguchi et al. 2021). For our method, we use a calibration dataset of 128 samples, each containing 2,048 tokens sampled from WikiText-2, unless otherwise specified. In our subspace alignment process, we apply expert grouping based on functional similarity using the expert output metric and K-means clustering as our default configuration. For expert merging, we employ frequency-based V matrix merging, which weighs by their activation frequency in the calibration data. We provide reproduced results of expert pruning methods, frequency-prune, output-prune based on frequency in MoE-

Model	Ratio	Runtime	ARC-c	ARC-e	BoolQ	HellaS.	MMLU	OBQA	RTE	WinoG.	Average↑
Mixtral 8×7B	0	87.7	0.56	0.84	0.85	0.65	0.67	0.35	0.71	0.76	0.67
Mixtral 6×7B	10%	93.1	0.44	0.75	0.78	0.52	0.52	0.31	0.62	0.72	0.58
	20%	104.7	0.38	0.70	0.67	0.46	0.43	0.28	0.58	0.68	0.52
	30%	120.9	0.29	0.60	0.63	0.38	0.33	0.22	0.53	0.61	0.45
Mixtral 4×7B	10%	95.3	0.40	0.70	0.72	0.48	0.41	0.28	0.56	0.69	0.53
	20%	108.2	0.34	0.65	0.65	0.43	0.37	0.23	0.53	0.65	0.48
	30%	122.7	0.26	0.55	0.62	0.36	0.29	0.19	0.53	0.61	0.43

Table 2: Performance of Sub-MoE under extra intra-expert compression ratios. Runtime denotes runtime throughput (Tokens/sec) on 8× H800 GPUs.

Compression (He et al. 2024) and expert merge (*i.e.*, MC-SMoE (Li et al. 2023b), HC-SMoE (Chen et al. 2024)). All experiments are conducted on eight NVIDIA H800 GPUs.

4.2 Performance Comparisons

Table 1 presents comprehensive comparisons of our Sub-MoE method against baseline approaches across four different MoE language models with varying degrees of expert reduction. The results demonstrate the consistent superiority of our proposed method across all evaluated models and compression ratios. For Mixtral-8×7B, when compressing from 8 to 6 experts, Sub-MoE achieves significantly better perplexity scores on WikiText-2 (5.16), PTB (18.58), and C4 (8.54) compared to pruning-based methods and other merging approaches. Notably, when reducing to just 4 experts (50% compression), our approach maintains impressive performance with an average accuracy of 0.58 across reasoning tasks, substantially outperforming the next best method HC-SMoE (0.51) and far surpassing pruning-based approaches that struggle to exceed 0.39 average accuracy. The performance gap still exists with the Qwen1.5-MoE-A2.7B-Chat model. Under extreme 50% compression from 60 to 30 experts, Sub-MoE achieves a robust average accuracy of 0.52. When examining larger models like Qwen3-30B-A3B with 128 experts, Sub-MoE demonstrates remarkable resilience even at 50% compression (64 experts), maintaining 0.57 average accuracy while HC-SMoE drops dramatically to 0.38. This pattern repeats with DeepSeek-MoE-16B, where our approach consistently preserves more of the original model's capabilities across multiple language tasks.

4.3 Effect of Intra-Expert Compression

Table 2 demonstrates the performance of Sub-MoE† under various intra-expert compression ratios across different model configurations. Our method exhibits remarkable robustness across all compression settings, maintaining reasonable performance even at aggressive compression rates. When compressing Mixtral 6x7B with a 10% ratio, Sub-MoE† achieves strong performance across downstream tasks, with an average score of 0.58 compared to the original model's 0.67, demonstrating effective parameter reduction while preserving model quality. The performance degrades gracefully as compression increases, with even 30% compression yielding usable results, highlighting Sub-MoE†'s superior compression efficiency and model preservation capabilities. As shown in Table 4, Sub-MoE† with fine-tuning (Sub-MoE†+FT) is able to additionally recover the accuracy significantly compared to other compression methods, achieving gains of 4-6% over the base Sub-MoE across benchmarks. Our method obtains stabilizing performance across diverse reasoning tasks, outperforming competing approaches (D^2-MoE (Gu et al. 2025b)), demonstrating robust generalization capabilities across various compression ratios and model sizes.

4.4 Ablation Study

Ablation on Core Components: Table 3 presents a comprehensive ablation study on key components of our Sub-MoE. **For the Clustering component (A),** we investigate three critical design choices: **(1) Multi-layer configuration in Adaptive Allocation** impacts performance, with 2-layer clustering (grouping 8×2=16 experts) yielding lowest perplexity and highest average accuracy. This balanced approach provides sufficient flexibility for identifying functional relationships while maintaining manageable cluster sizes. In contrast, 1-layer clustering limits the diversity of potential merge candidates, while 3-layer clustering creates overly complex groupings that lead to accuracy drops. **(2) Similarity Metric** comparison reveals that while router-logits and weight-based similarity measures perform reasonably well, our expert output similarity metric achieves the best overall balance between language modeling and reasoning tasks (0.64 mean accuracy). **(3) Clustering Algorithm** analysis shows that K-means consistently delivers optimal results compared to random grouping or hierarchical clustering, particularly on language modeling tasks.

For Merging component (B), we examine two key aspects: **(5)** U**-Sharing strategy** comparison demonstrates that our union SVD approach substantially outperforms vanilla SVD across all metrics (8.54 vs. 10.25 on C4; 0.64 vs. 0.58 average accuracy), with particularly notable improvements on MMLU (0.59 vs. 0.49). This confirms the effectiveness of our approach in finding a common representational space that preserves expert functionality. **(6)** V**-Merging strategy** experiments show that our frequency-based approach consistently outperforms both dropping the least significant components and simple averaging. The frequency-weighted approach maintains better overall performance, demonstrating the importance of respecting expert utilization patterns

Settings	Options	WikiText-2↓	PTB↓	C4↓	ARC-c	ARC-e	BoolQ	HellaS.	MMLU	OBQA	RTE	WinoG.	Average↑
(A) Adaptive Expert Clustering Settings													
Clustering Layer	Sub-MoE (1-Layer)	5.32	20.08	8.77	0.48	0.79	0.78	0.61	0.59	0.30	0.63	0.75	0.62
	Sub-MoE (2-Layer)	**5.16**	**18.58**	**8.54**	**0.49**	**0.80**	**0.86**	**0.62**	**0.59**	**0.32**	**0.65**	**0.75**	**0.64**
	Sub-MoE (3-Layer)	6.03	21.17	9.52	0.47	0.76	0.84	0.59	0.31	0.30	0.59	0.73	0.57
Similarity Metric	Sub-MoE (Router-logits)	5.75	19.16	9.19	0.48	0.78	0.80	0.60	0.57	0.29	0.65	0.74	0.61
	Sub-MoE (Weight)	6.41	19.46	10.09	0.44	0.7513	0.84	0.57	0.52	0.30	0.55	0.72	0.59
	Sub-MoE (Expert output)	**5.16**	**18.58**	**8.54**	**0.49**	**0.80**	**0.86**	**0.62**	**0.59**	**0.32**	**0.65**	**0.75**	**0.64**
Clustering Alg.	Sub-MoE (Random)	6.12	18.86	9.63	0.46	0.76	0.84	0.59	0.52	0.30	0.61	0.73	0.61
	Sub-MoE (Hierarchical)	5.46	19.30	9.01	0.50	0.73	0.82	0.61	0.62	0.33	0.69	0.71	0.63
	Sub-MoE (K-means)	**5.16**	**18.58**	**8.54**	**0.49**	**0.80**	**0.86**	**0.62**	**0.59**	**0.32**	**0.65**	**0.75**	**0.64**
(B) Subspace Expert Merging Settings													
U-Sharing	Sub-MoE (Vanilla SVD)	6.57	20.72	10.25	0.43	0.75	0.82	0.57	0.49	0.28	0.61	0.73	0.58
	Sub-MoE (Union SVD)	**5.16**	**18.58**	**8.54**	**0.49**	**0.80**	**0.86**	**0.62**	**0.59**	**0.32**	**0.65**	**0.75**	**0.64**
V-Merging	Sub-MoE (Drop)	5.53	19.77	9.05	0.50	0.80	0.84	0.59	0.59	0.32	0.61	0.71	0.61
	Sub-MoE (Average)	5.31	18.63	8.88	0.50	0.81	0.85	0.61	0.59	0.31	0.64	0.74	0.62
	Sub-MoE (Frequency)	**5.16**	**18.58**	**8.54**	**0.49**	**0.80**	**0.86**	**0.62**	**0.59**	**0.32**	**0.65**	**0.75**	**0.64**

Table 3: Ablation on our (A) Expert Clustering and (B) Subspace Merging for Mixtral 8x7B→6x7B.

Method	**ARC-e**	**WinoG.**	**ARC-c**
SVD	0.30	0.52	0.22
ASVD (Yuan et al. 2023)	0.41	0.58	0.22
SVD-LLM (Wang et al. 2024)	0.43	0.52	0.22
NAEE (Lu et al. 2024)	0.63	0.64	0.36
MoE-I^2 (Yang et al. 2024b),	0.68	0.66	0.38
D^2-MoE (Gu et al. 2025b)	0.68	0.63	0.37
MoE-SVD (Li et al. 2025)	0.66	0.67	0.34
MC-SMoE (Li et al. 2023b)	0.54	0.60	0.26
Sub-MoE†	**0.70**	**0.68**	**0.38**
Sub-MoE†+FT	**0.74**	**0.71**	**0.44**

Table 4: Performance comparison of more compression methods on reasoning benchmarks.

Metric	**Samples**				
	32	64	96	128	256
WikiText-2 Perplexity	8.69	8.11	7.32	5.16	5.02
C4 Perplexity	23.89	19.98	15.68	8.54	8.23

Table 5: Perplexity values for WikiText-2 and C4 datasets across different sample sizes.

when merging. These ablation results empirically validate our design choices and demonstrate that each component of Sub-MoE contributes meaningfully to its overall effectiveness.

Impact of Calibration Size . Table 5 shows how calibration sample size affects model performance. Increasing samples from 32 to 128 substantially reduces perplexity on WikiText-2 (8.69→5.16) and C4 (23.89→8.54), while further increases yield minimal gains.

Memory and Runtime Analysis. As shown in Table 6, reducing Mixtral-8×7B from 8 to 6 experts decreases memory by 24% with only a 6% accuracy drop, while compression to 4 experts achieves optimal efficiency with 48% memory reduction and 13% accuracy decline. Compression below 4 experts causes disproportionate performance degradation, indicating a practical lower bound for maintaining capabilities. For runtime, our Sub-MoE† can around 1.3× throughput speedup (see Table 2).

Metric	**3 Experts**	**4 Experts**	**6 Experts**	**8 Experts**
Memory Usage (GB)	34.58	45.49	66.49	87.49
Mean Accuracy	0.55	0.58	0.63	0.67

Table 6: Memory usage and mean accuracy for different numbers of experts (horizontal layout).

5 Conclusions

In this paper, we present Sub-MoE, a new expert merging framework that addresses parameter conflicts in MoE LLM compression through subspace alignment. By decomposing concatenated experts via SVD, our approach extracts shared U-matrices while enabling the effective merging of expert-specific V components. Our two-phase, Adaptive Expert Clustering and Subspace Expert Merging, identifies functionally similar experts and combines them with minimal information loss. Extensive experiments on Mixtral, DeepSeek, and Qwen MoE LLMs reveal that our approach consistently outperforms state-of-the-art pruning and merging baselines. In future work, we will extend Sub-MoE with other compression schemes (AutoML (Li et al. 2024d,e), distillation (Li 2022; Li and Jin 2022; Li et al. 2023a, 2024a,b; Dong, Li, and Wei 2023), quantization (Dong et al. 2025; Gu et al. 2025a), sparsity (Li et al. 2024c,f)) and explore calibration-free ways (Ghaffari et al. 2024).

Acknowledgments

This work is funded in part by the HKUST Start-up Fund (R9911), Theme-based Research Scheme grant (T45-205/21-N), the InnoHK funding for Hong Kong Generative AI Research and Development Center, Hong Kong SAR, and the research funding under HKUST-DXM AI for Finance Joint Laboratory (DXM25EG01).

References

Bentivogli, L.; Clark, P.; Dagan, I.; and Giampiccolo, D. 2009. The Fifth PASCAL Recognizing Textual Entailment Challenge. *TAC*, 7(8): 1.

Cai, W.; Jiang, J.; Qin, L.; Cui, J.; Kim, S.; and Huang, J. 2024. Shortcut-connected Expert Parallelism for Accelerating Mixture-of-Experts. *arXiv preprint arXiv:2404.05019*.

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*.

Chen, T.; Huang, S.; Xie, Y.; Jiao, B.; Jiang, D.; Zhou, H.; Li, J.; and Wei, F. 2022. Task-specific expert pruning for sparse mixture-of-experts. *arXiv preprint arXiv:2206.00277*.

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. arXiv:1905.10044.

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:1803.05457*.

Dai, D.; Deng, C.; Zhao, C.; Xu, R.; Gao, H.; Chen, D.; Li, J.; Zeng, W.; Yu, X.; Wu, Y.; et al. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*.

DeepSeek-AI; et al. 2024. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. arXiv:2405.04434.

Dong, P.; Li, L.; and Wei, Z. 2023. DisWOT: Student Architecture Search for Distillation WithOut Training. In *CVPR*.

Dong, P.; Li, L.; Zhong, Y.; Du, D.; Fan, R.; Chen, Y.; Tang, Z.; Wang, Q.; Xue, W.; Guo, Y.; et al. 2025. STBLLM: Breaking the 1-Bit Barrier with Structured Binary LLMs. In *ICLR*.

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. 2023. A framework for few-shot language model evaluation.

Ghaffari, A.; Younesian, S.; Nia, V. P.; Chen, B.; and Asgharian, M. 2024. AdpQ: A Zero-shot Calibration Free Adaptive Post Training Quantization Method for LLMs. *arXiv preprint arXiv:2405.13358*.

Gu, H.; Li, L.; Wang, Z.; Liu, B.; Zhu, Q.; Han, S.; and Guo, Y. 2025a. BTC-LLM: Efficient Sub-1-Bit LLM Quantization via Learnable Transformation and Binary Codebook. *arXiv preprint arXiv:2506.12040*.

Gu, H.; Li, W.; Li, L.; Qiyuan, Z.; Lee, M.; Sun, S.; Xue, W.; and Guo, Y. 2025b. Delta Decompression for MoE-based LLMs Compression. In *Forty-second International Conference on Machine Learning*.

He, S.; Dong, D.; Ding, L.; and Li, A. 2024. Demystifying the Compression of Mixture-of-Experts Through a Unified Framework. *arXiv preprint arXiv:2406.02500*.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*.

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 *The Thirteenth International Conference on Learning Representations*.

Ikotun, A. M.; Ezugwu, A. E.; Abualigah, L.; Abuhaija, B.; and Heming, J. 2023. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 622: 178–210.

Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.; and Wilson, A. G. 2018. Averaging weights leads to wider optima and better generalization. *UAI*.

Jiang, A. Q.; Sablayrolles, A.; Roux, A.; Mensch, A.; Savary, B.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Hanna, E. B.; Bressand, F.; Lengyel, G.; Bour, G.; Lample, G.; Lavaud, L. R.; Saulnier, L.; Lachaux, M.-A.; Stock, P.; Subramanian, S.; Yang, S.; Antoniak, S.; Scao, T. L.; Gervet, T.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2024a. Mixtral of Experts. arXiv:2401.04088.

Jiang, C.; Tian, Y.; Jia, Z.; Zheng, S.; Wu, C.; and Wang, Y. 2024b. Lancet: Accelerating Mixture-of-Experts Training via Whole Graph Computation-Communication Overlapping. *arXiv preprint arXiv:2404.19429*.

Lee, J.; Qiao, A.; Campos, D. F.; Yao, Z.; He, Y.; et al. 2024. STUN: Structured-Then-Unstructured Pruning for Scalable MoE Pruning. *arXiv preprint arXiv:2409.06211*.

Li, L. 2022. Self-Regulated Feature Learning via Teacher-free Feature Distillation. In *ECCV*.

Li, L.; Bao, Y.; Dong, P.; Yang, C.; Li, A.; Luo, W.; Liu, Q.; Xue, W.; and Guo, Y. 2024a. DetKDS: Knowledge Distillation Search for Object Detectors. In *ICML*.

Li, L.; Dong, P.; Li, A.; Wei, Z.; and Yang, Y. 2024b. Kd-zero: Evolving knowledge distiller for any teacher-student pairs. *NeuIPS*.

Li, L.; Dong, P.; Wei, Z.; and Yang, Y. 2023a. Automated knowledge distillation via monte carlo tree search. In *ICCV*.

Li, L.; and Jin, Z. 2022. Shadow Knowledge Distillation: Bridging Offline and Online Knowledge Transfer. In *NeuIPS*.

Li, L.; Peijie; Tang, Z.; Liu, X.; Wang, Q.; Luo, W.; Xue, W.; Liu, Q.; Chu, X.; and Guo, Y. 2024c. Discovering Sparsity Allocation for Layer-wise Pruning of Large Language Models. In *NeuIPS*.

Li, L.; Sun, H.; Li, S.; Dong, P.; Luo, W.; Xue, W.; Liu, Q.; and Guo, Y. 2024d. Auto-gas: Automated proxy discovery for training-free generative architecture search. ECCV.

Li, L.; Wei, Z.; Dong, P.; Luo, W.; Xue, W.; Liu, Q.; and Guo, Y. 2024e. Attnzero: efficient attention discovery for vision transformers. In *ECCV*.

Li, M.; Gururangan, S.; Dettmers, T.; Lewis, M.; Althoff, T.; Smith, N. A.; and Zettlemoyer, L. 2022. Branch-Train-Merge: Embarrassingly Parallel Training of Expert Language Models. In *First Workshop on Interpolation Regularizers and Beyond at NeurIPS 2022*.

Li, P.; Zhang, Z.; Yadav, P.; Sung, Y.-L.; Cheng, Y.; Bansal, M.; and Chen, T. 2023b. Merge, then compress: Demystify efficient SMoe with hints from its routing policy. *arXiv preprint arXiv:2310.01334*.

Li, W.; Li, L.; Huang, Y.-L.; Lee, M. G.; Sun, S.; Xue, W.; and Guo, Y. 2025. Structured Mixture-of-Experts LLMs Compression via Singular Value Decomposition. In *ICML*.

Li, W.; Li, L.; Lee, M.; and Sun, S. 2024f. ALS: Adaptive Layer Sparsity for Large Language Models via Activation Correlation Assessment. In *NeuIPS*.

Liu, E.; Zhu, J.; Lin, Z.; Ning, X.; Blaschko, M. B.; Yan, S.; Dai, G.; Yang, H.; and Wang, Y. 2024. Efficient expert pruning for sparse mixture-of-experts language models: Enhancing performance and reducing inference costs. *arXiv preprint arXiv:2407.00945*.

Liu, M.; Wang, W.; and Wu, C. 2025. Optimizing Distributed Deployment of Mixture-of-Experts Model Inference in Serverless Computing. *arXiv preprint arXiv:2501.05313*.

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 *ACL*.

Matena, M.; and Raffel, C. 2022. Merging models with fisher-weighted averaging. In *NeurIPS*.

Merity, S.; Xiong, C.; Bradbury, J.; and Socher, R. 2016. Pointer Sentinel Mixture Models. arXiv:1609.07843.

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*, 2381–2391.

Mikolov, T.; Karafiát, M.; Burget, L.; Cernockỳ, J.; and Khudanpur, S. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, 1045–1048. Makuhari.

Muzio, A.; Sun, A.; and He, C. 2024. SEER-MoE: Sparse Expert Efficiency through Regularization for Mixture-of-Experts. *arXiv preprint arXiv:2404.05089*.

Nguyen, N. V.; Doan, T. T.; Tran, L.; Nguyen, V.; and Pham, Q. 2024. LIBMoE: A Library for comprehensive benchmarking Mixture of Experts in Large Language Models. *arXiv preprint arXiv:2411.00918*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2023. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv:1910.10683.

Sakaguchi, K.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2021. WinoGrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9): 99–106.

Sarkar, S.; Lausen, L.; Cevher, V.; Zha, S.; Brox, T.; and Karypis, G. 2024. Revisiting SMoE Language Models by Evaluating Inefficiencies with Task Specific Expert Pruning. *arXiv preprint arXiv:2409.01483*.

Shen, L.; Wu, Z.; Gong, W.; Hao, H.; Bai, Y.; Wu, H.; Wu, X.; Bian, J.; Xiong, H.; Yu, D.; et al. 2022. Se-moe: A scalable and efficient mixture-of-experts distributed training and inference system. *arXiv preprint arXiv:2205.10034*.

Song, Y.; Mi, Z.; Xie, H.; and Chen, H. 2023. PowerInfer: Fast Large Language Model Serving with a Consumer-grade GPU. arXiv:2312.12456.

Team, Q. 2024. Qwen1.5-MoE: Matching 7B Model Performance with 1/3 Activated Parameters".

Wang, X.; Zheng, Y.; Wan, Z.; and Zhang, M. 2024. Svd-llm: Truncation-aware singular value decomposition for large language model compression. *arXiv preprint arXiv:2403.07378*.

Wortsman, M.; Ilharco, G.; Gadre, S. Y.; Roelofs, R.; Gontijo-Lopes, R.; Morcos, A. S.; Namkoong, H.; Farhadi, A.; Carmon, Y.; Kornblith, S.; et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *ICML*.

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*.

Xue, L.; Fu, Y.; Lu, Z.; Mai, L.; and Marina, M. 2024. Moe-infinity: Activation-aware expert offloading for efficient moe serving. *arXiv preprint arXiv:2401.14361*.

Yadav, P.; Tam, D.; Choshen, L.; Raffel, C.; and Bansal, M. 2023. TIES-Merging: Resolving Interference When Merging Models. arXiv:2306.01708.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Lin, J.; Dang, K.; Lu, K.; Bao, K.; Yang, K.; Yu, L.; Li, M.; Xue, M.; Zhang, P.; Zhu, Q.; Men, R.; Lin, R.; Li, T.; Xia, T.; Ren, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Qiu, Z. 2024a. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*.

Yang, C.; Sui, Y.; Xiao, J.; Huang, L.; Gong, Y.; Duan, Y.; Jia, W.; Yin, M.; Cheng, Y.; and Yuan, B. 2024b. MoE-I2: Compressing Mixture of Experts Models through Inter-Expert Pruning and Intra-Expert Low-Rank Decomposition. *arXiv preprint arXiv:2411.01016*.

Yuan, Z.; Shang, Y.; Song, Y.; Wu, Q.; Yan, Y.; and Sun, G. 2023. ASVD: Activation-aware Singular Value Decomposition for Compressing Large Language Models. *CoRR*, abs/2312.05821.

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? arXiv:1905.07830.

Zhao, H.; Qiu, Z.; Wu, H.; Wang, Z.; He, Z.; and Fu, J. 2024. HyperMoE: Towards Better Mixture of Experts via Transferring Among Experts. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *ACL*.