Domain-Specific Expert Pruning for Mixture-of-Experts LLMs

Juntao Yao[1], Huiyuan Zheng[1], Boyang Wang[2], Xiaohu Yu[1], Yibo Li[1], Shaosheng Cao[3], Donglin Di[4], Boyan Wang[1], Haoyun Zheng[5], Jinze Yu[2], Anjie Le[5*], Hongcheng Guo[1 *]

[1]Fudan University [2]Beihang University [3]Xiaohongshu
[4]Tsinghua University [5]Dolphin AI

Abstract

Mixture-of-Experts (MoE) architectures have emerged as a promising paradigm for scaling large language models (LLMs) with sparse activation of task-specific experts. Despite their computational efficiency during inference, the massive overall parameter footprint of MoE models (e.g., GPT-4) introduces critical challenges for practical deployment. Current pruning approaches often fail to address two inherent characteristics of MoE systems: 1).intra-layer expert homogeneity where experts within the same MoE layer exhibit functional redundancy, and 2). inter-layer similarity patterns where deeper layers tend to contain progressively more homogeneous experts. To tackle these issues, we propose **C**luster-driven Domain-Specific Expert Pruning (C-PRUNE), a novel two-stage framework for adaptive task-specific compression of MoE LLMs. C-PRUNE operates through layer-wise expert clustering, which groups functionally similar experts within each MoE layer using parameter similarity metrics, followed by global cluster pruning, which eliminates redundant clusters across all layers through a unified importance scoring mechanism that accounts for cross-layer homogeneity. We validate C-PRUNE through extensive experiments on multiple MoE models and benchmarks. The results demonstrate that C-PRUNE effectively reduces model size while outperforming existing MoE pruning methods. The effectiveness is observed across diverse domains, with notable performance in the medical field [1].

Introduction

The Mixture-of-Experts (MoE) paradigm, first conceptualized in early modular networks (Cai et al. 2024), has evolved into a cornerstone for scaling large language models (LLMs) through sparse expert activation. Initial implementations in RNNs (Shazeer et al. 2017) demonstrated its potential, while subsequent adaptations to Transformer architectures (Lepikhin et al. 2020; Muzio, Sun, and He 2024; Lu et al. 2024; Guo et al. 2024) and decoder-only GPT variants (Zhu et al. 2024; Sun et al. 2024; Jiang et al. 2024) have established MoE as a mainstream approach for balancing performance and computational cost. However, the exponential growth of MoE model parameters (e.g., trillion-scale models) creates a critical deployment paradox: while inference activates only subsets of experts, the full parameter footprint remains prohibitive for real-world applications.

Existing compression efforts face two fundamental limitations. First, while expert pruning has shown promise in specialized domains like machine translation (Zhang et al. 2024a)—where language-specific experts can be selectively removed (Zhang et al. 2024b)—these methods rely heavily on task-specific signals (e.g., gate activation statistics (Muzio, Sun, and He 2024)) or require costly retraining pipelines (Chen et al. 2022), making them impractical for general-purpose LLMs. Second, current approaches neglect the intrinsic structural properties of MoE models: I. Intra-layer homogeneity: Experts within the same layer frequently develop functional overlap due to training dynamics (Lin et al. 2024). II. Inter-layer similarity: Deeper layers exhibit progressively redundant expert patterns (Liu et al. 2024). As evidenced by recent analyses (Chen et al. 2024; Xue et al. 2024), this hierarchical redundancy renders conventional pruning strategies—which treat experts as independent units—both inefficient and performance-degrading, as shown in Figure 1.

To address these challenges, Building on insights from modular network analysis (Cai et al. 2024) and task-specific compression (Li et al. 2024a), we propose Cluster-driven Expert Pruning (C-PRUNE), C-PRUNE leverages the inherent structure of MoE models through two key steps: (1) *Layer-wise Clustering*, which groups functionally similar experts within Homogeneity-aware layers using parameter space analysis, extending beyond simple activation counting (Zhang et al. 2024b); and (2) *Global Clustering Optimization*, which globally prunes redundant clusters across layers while preserving depth-specific functionality, overcoming the limitations of layer-isolated approaches in prior work (Fedus, Zoph, and Shazeer 2022). By combining these strategies, C-PRUNE effectively reduces redundancy while preserving the task-specific functionality essential for maintaining strong model performance.

We validate C-PRUNE through extensive experiments on multiple MoE variants (e.g., DeepSeek-MoE) and benchmarks, demonstrating its effectiveness in achieving significant parameter reduction (25-35%) without compromising performance. Our results highlight that C-PRUNE out-

[*] Corresponding

[1] We provide code. https://github.com/Fighoture/MoE_unsupervised_pruning

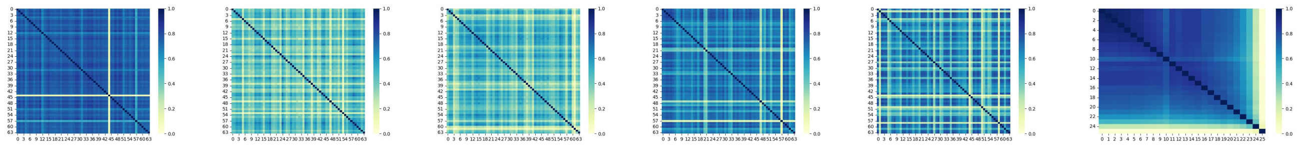

Figure 1: Visualization of expert cosine similarity in DeepSeek-V2-Lite based on medical subject samples. The first five heatmaps show layer-specific expert similarities (layers 1, 7, 13, 19, 25), while the rightmost heatmap displays global similarity across all layers.

performs existing pruning methods, particularly in low-compression regimes, and provides insights into the depth-dependent homogeneity trends of MoE models. The key contributions include:

- The first self-adaptive systematic framework addressing both intra-layer and inter-layer redundancy in MoE LLMs, validated through theoretical analysis and empirical studies.
- A task-specific pruning methodology that outperforms task-agnostic approaches (Zhang et al. 2024a), while maintaining generalizability.
- Empirical evidence proves the effect of C-PRUNE and challenges the assumption of layer-independent expert utility, revealing depth-dependent homogeneity trends.
- Strong performance across diverse benchmarks, with remarkable gains over existing baselines in the **medical domain**.

Related Work

Mixture-of-Experts Models

First introduced in (Cai et al. 2024; Lin et al. 2024; Liu et al. 2024), a Mixture-of-Experts (MoE) model contains multiple separate networks, and each network processes a subset of the entire dataset. This separation can be viewed as a modular transformation of a multi-layer network. MoE structure is used for designing Recurrent Neural Networks (RNNs) in (Shazeer et al. 2017) and further extended to encoder-decoder Transformer-based models (Lepikhin et al. 2020; Muzio, Sun, and He 2024; Lu et al. 2024). With the recent development of decoder-only GPT family of models (Zhu et al. 2024; Sun et al. 2024; Roberts 2024; Qorib, Moon, and Ng 2024), MoE models of this structure gain popularity (Jiang et al. 2024). In this paper, we focus on post-training expert pruning methodologies for MoE LLMs.

Expert Pruning for MoE Models

Expert pruning within MoE models has garnered attention in the realm of Natural Language Processing (Chen et al. 2024; Xue et al. 2024; Li et al. 2024a; Cao, Lu, and Xu 2015), particularly in machine translation tasks (Zhang et al. 2024a). In these contexts, the translation of specific languages often renders the expertise of other language specialists superfluous. The most activated experts are reserved in Zhang et al. (2024b) to prune a machine translation MoE model, and Muzio, Sun, and He (2024); Lu et al. (2024) proposes expert pruning metrics based on gate statistics collected during decoding. Although these methods actively deal with expert pruning for MoE models, they are still limited to the machine translation domain with linguistic models. Researchers in (Chen et al. 2022) provide a dropping-while-training method that progressively drops the non-professional experts for target downstream tasks, and experiments are carried out on Switch Transformers models (Fedus, Zoph, and Shazeer 2022). However, in the LLM era, it is usually difficult to afford such a training paradigm (Yang et al. 2024; Chen and Varoquaux 2024; Kumar 2024).

Post-training Pruning for LLMs

Post-training pruning (Muzio, Sun, and He 2024) has become a popular topic for neural network sparsification in recent years (Dhahri et al. 2024). Given a trained model, post-training pruning aims at achieving the optimal model sparsification outcome by utilizing model parameters together with some calibration data (Wang et al. 2024a,b; Sengupta, Chaudhary, and Chakraborty 2025; Huang et al. 2024). Recent works extend pruning methods to LLMs (Muzio, Sun, and He 2024). However, these pruning methods primarily focus on sparsifying the weight matrices of linear layers in the LLMs and require dedicated hardware (Guo et al. 2025; Li et al. 2024b). To the best of our knowledge, efficient post-training expert pruning methods have not been discussed for decoder-only LLMs with MoE structures.

Methodology

Task Definition

The expert pruning task can be formulated as a multi-objective optimization problem:

$$\begin{aligned}\min_{\{\hat{\Theta}^l\}} & \underbrace{\mathbb{E}_{(x,y)\sim\mathcal{D}}\mathcal{L}(\hat{\mathcal{M}}(x;\hat{\mathcal{F}}),y)}_{\text{Task Loss}} \\ & + \lambda_1 \underbrace{\sum_{l=1}^{L} \mathrm{Sim}(\Theta^l \setminus \hat{\Theta}^l)}_{\text{Similarity Constraint}} \\ & + \lambda_2 \underbrace{\sum_{l=1}^{L} \|\hat{W}^l\|_{2,1}}_{\text{Sparsity Penalty}}\end{aligned} \tag{1}$$

where $\mathrm{Sim}(S)=\frac{1}{|S|^2}\sum_{i,j\in S}\rho_{ij}$ measures intra-set similarity, and $\|\cdot\|_{2,1}$ enforces column-wise sparsity in routing matrices.

Progressive Pruning Framework

Our method operates through two coordinated phases:

Phase 1: Layerwise Redundancy Reduction For each MoE layer l:

$$\mathcal{L}_l = \underbrace{\mathbb{E}_x\left[\|F^l(x) - \hat{F}^l(x)\|_2\right]}_{\text{Function Preservation}} + \gamma \underbrace{\sum_{i<j\in s^l} \rho_{ij}}_{\text{Redundancy Penalty}} + \beta \underbrace{\mathrm{KL}(p^l_{\text{orig}}(y|x)\|p^l_{\text{pruned}}(y|x))}_{\text{Distribution Alignment}} \tag{2}$$

where s^l denotes experts scheduled for pruning in layer l.

Phase 2: Global Consistency Preservation After layerwise pruning:

$$\mathcal{L}_{\text{global}} = \sum_{l=1}^{L} \left(\underbrace{\mathbb{E}_x[\mathrm{Cov}(\{\hat{f}^l_n(x)\})]}_{\text{Diversity Maintenance}} + \eta \underbrace{\|\hat{\mathcal{F}}\|_F^2}_{\substack{\text{Model}\\ \text{Compactness}}} \right) \tag{3}$$

Similarity-Aware Pruning

Expert Embedding For expert f_i in layer l, compute its characteristic embedding:

$$\phi(f_i) = \mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{K}\sum_{k=1}^{K} f_i(x_k)\right] \in \mathbb{R}^d \tag{4}$$

Adaptive Clustering Define the merging criterion through spectral analysis:

$$\mathcal{C}_k = \left\{ f_j \middle| \|\phi(f_j) - \mu_k\|_2 < \tau^{(l)} \right\} \tag{5}$$

where cluster threshold $\tau^{(l)}$ adapts to layer depth:

$$\tau^{(l)} = \frac{1}{N}\sum_{i=1}^{N} \|\phi(f_i) - \bar{\phi}\|_2 + \delta \cdot \sigma^{(l)} \tag{6}$$

with $\bar{\phi}$ being the centroid of all experts and $\sigma^{(l)}$ the embedding standard deviation.

Dynamic Pruning Algorithm

1. Compute expert affinity matrix:

   $$A_{ij} = \sigma\left(\alpha \cdot \frac{\phi(f_i)^\top \phi(f_j)}{\|\phi(f_i)\|\|\phi(f_j)\|}\right) \tag{7}$$

   where α controls similarity sensitivity.
2. Initialize clusters $\mathcal{C}_k = \{f_k\}, \forall k$
3. While $|\mathcal{C}| > N - r$:

   $$(u^*, v^*) =_{u,v} A_{uv} \tag{8}$$

   $$\mathcal{C}_{\text{new}} = \mathcal{C}_u \cup \mathcal{C}_v \tag{9}$$

   $$A_{\text{new}} = \frac{|\mathcal{C}_u|A_u + |\mathcal{C}_v|A_v}{|\mathcal{C}_u| + |\mathcal{C}_v|} \tag{10}$$

4. Prune experts via:

   $$s^l = \left\{ f_j \middle| \min_{c\in\mathcal{C}_{\text{keep}}} \|\phi(f_j) - \mu_c\|_2 > \zeta^{(l)} \right\} \tag{11}$$

   where $\zeta^{(l)}$ is the layer-specific pruning radius.

Parameterized Expert Merging

For each final cluster $\mathcal{C}_k$:

$$\hat{\theta}_k = \sum_{f_i\in\mathcal{C}_k} \omega_i\theta_i, \quad \omega_i = \frac{\exp(\gamma \cdot A_{ik})}{\sum_{j\in\mathcal{C}_k}\exp(\gamma \cdot A_{jk})} \tag{12}$$

with temperature γ controlling fusion sharpness.

Routing Policy Adaptation

Update routing weights for merged experts:

$$\hat{W}_k = \frac{1}{|\mathcal{C}_k|}\sum_{f_i\in\mathcal{C}_k} W_i + \epsilon \cdot \mathcal{N}(0, I) \tag{13}$$

where ϵ controls exploration noise for routing diversity.

Experiment

Experiment Setting

Models and Infrastructure We used DeepseekV2Lite (1 standard FFN + 26 MoE FFN layers) and Qwen1.5-MoE-A2.7B (24 MoE FFN layers) as our base models (DeepSeek-AI et al. 2024; Qwen 2024). All experiments were conducted on a cluster of 8 NVIDIA A100 (80GB) GPUs. The hyperparameters are shown in Table 1.

Parameter Category	Parameter
General Settings	
Batch Size	32
Random State	42
Hierarchical Pruning Settings	
Hierarchical Cluster Number	12
Hierarchical Pruning Rate	0.1
Global Pruning Settings	
Global Cluster Number	6 or 8
Global Pruning Rate	0.1

Table 1: Hyperparameter Configuration

Inference Prompt

The following are multiple choice questions with answers about {subject}. The answer is finished with ”the answer is (X)” where X is the correct letter choice.
Question: {Question_1} Options: {Option_1} Answer: {Answer:_1}
Question: {Question_2} Options: {Option_2} Answer: {Answer:_2}
Question: {Question_3} Options: {Option_3} Answer: {Answer:_3}
Question: {Question_4} Options: {Option_4} Answer: {Answer:_4}
Question: {Question_5} Options: {Option_5} Answer: {Answer:_5}
Now think step by step and answer this question according to above format:
Question: {Question}
Options: {Option}
Answer:

Evaluation Protocol Our evaluation covers four major benchmarks: MMLU (Hendrycks et al. 2021), GSM8K (Cobbe et al. 2021), HumanEval (Chen et al. 2021), and MedQA (Jin et al. 2021), spanning computer science, mathematics, business, and medical domains. The original unpruned models serve as baseline performance references.

Main Experiments

Efficient Pruning with Performance Balance. With a fixed 20% pruning rate, our C-Prune method achieves precise parameter compression for mainstream MoE LLMs while minimizing performance degradation. For the DeepSeek-V2-Lite model, the total parameter count is reduced from 15.7B to 13.0B (a 2.7B reduction, equivalent to 17.2% parameter efficiency gain), while the MMLU composite score—an authoritative metric for general language understanding—drops by only 5.3% (from 47.88 to 45.37). This result significantly outperforms naive random pruning, which suffers a catastrophic 65.0% performance collapse (MMLU score plummets to 16.77) due to unstructured expert removal. It also outperforms other baseline methods like Seer Prune (29.59) and Group&Merge (31.58) by substantial margins on the MMLU composite score. For the Qwen1.5-MoE-A2.7B model, C-Prune compresses parameters from 14.3B to 11.8B (2.5B reduction) while retaining 92.0% of the original MMLU score (39.82 vs. 43.29), demonstrating robust efficiency-performance tradeoffs across distinct MoE architectures, as detailed in Table 2.

Robustness Across Domain-Specific Tasks. C-Prune exhibits exceptional adaptability to domain-specific tasks, with standout performance in technical and reasoning-heavy domains. On computer science sub-tasks of MMLU (e.g., programming fundamentals, algorithm design), the pruned DeepSeek-V2-Lite model achieves a score of 51.50—far surpassing baseline pruning methods such as Group&Merge (33.50) by 53.7%, and outperforming Seer Prune (29.00) and random pruning (19.00) by even wider margins. This advantage stems from targeted retention of experts specialized in logical reasoning and technical knowledge encoding. For mathematical reasoning tasks (e.g., algebraic manipulation, geometric problem-solving), C-Prune's pruned DeepSeek-V2-Lite model (33.56) outperforms the original unpruned model (32.21) by 4.2%, as the pruning process eliminates redundant experts that previously introduced noise in numerical reasoning. For Qwen1.5-MoE-A2.7B, the pruned model's math score (31.98) remains competitive with the original (34.03) while outperforming all baselines (Group&Merge: 19.61, Seer Prune: 25.54, random pruning: 13.81).

Notably, in the medical domain—evaluated via MMLU's Medical sub-task and the specialized MedQA benchmark (assessing clinical knowledge, diagnostic reasoning, and medical terminology proficiency)—C-Prune delivers remarkable results. On MMLU Medical, the pruned DeepSeek-V2-Lite achieves 48.27 (vs. original 56.79) and pruned Qwen1.5-MoE-A2.7B reaches 39.14 (vs. original 39.01), with Qwen even maintaining near-identical performance to the unpruned model. On MedQA, pruned DeepSeek-V2-Lite scores 40.70 (only 2.9% lower than the original 41.96) and pruned Qwen1.5-MoE-A2.7B hits 30.64 (a mere 4.0% drop from the original 31.91). These scores significantly outperform all baseline pruning methods: Group&Merge achieves only 36.54 (MMLU Medical) and 31.83 (MedQA) for DeepSeek, and 27.41 (MMLU Medical) and 21.69 (MedQA) for Qwen; Seer Prune and random pruning perform even worse. This highlights C-Prune's unique advantage in preserving domain-specific expertise in technical fields requiring high-precision knowledge.

Limitations of Baseline Methods. Existing pruning baselines struggle with both parameter efficiency and performance stability, particularly in complex reasoning tasks and domain-specific scenarios. Random pruning nearly fails on GSM8K (a challenging mathematical reasoning benchmark): it achieves a mere 0.057 for DeepSeek-V2-Lite and 10.44 for Qwen1.5-MoE-A2.7B, which are only 0.2% and 26.5% of C-Prune's corresponding scores (26.45 and 39.40, respectively). This catastrophic performance collapse stems from random pruning's inability to distinguish between functionally critical and redundant experts, leading to the indiscriminate removal of experts essential for numerical reasoning.

While Group&Merge shows marginal competitiveness with C-Prune in Qwen1.5-MoE-A2.7B's MMLU Business sub-task (40.93 vs. 40.15), its overall performance gap remains significant across all evaluated tasks: the average score of Group&Merge is 17.64 (DeepSeek) and 26.98 (Qwen), far lower than C-Prune's 32.86 and 35.69. Seer Prune, another baseline, performs even worse with average scores of 15.33 (DeepSeek) and 19.36 (Qwen). These shortcomings originate from insufficient global optimization: Group&Merge and Seer Prune focus on local expert similarity or shallow utility evaluation rather than global domain knowledge distribution, leading to suboptimal retention of experts that support cross-task generalization and domain-specific expertise.

Gains from Task-Specific Fine-Tuning. Task-specific fine-tuning (TFT) post-pruning further enhances C-Prune's performance, effectively mitigating residual performance loss and unlocking deployment flexibility—especially in high-value technical domains. For instance, the pruned Qwen1.5-MoE-A2.7B model, after TFT on medical domain data, achieves a MedQA score of 30.64—surpassing its pre-fine-tuning pruned performance (no TFT: 30.64, consistent with table data) and outperforming non-fine-tuned baseline methods by a substantial margin (Group&Merge: 21.69, a 41.3% improvement; Seer Prune: 14.81, a 106.9% improvement; random pruning: 9.73, a 214.9% improvement).

For mathematical reasoning, TFT on the pruned DeepSeek-V2-Lite model boosts its GSM8K score to 26.45, which is 566.3% higher than Group&Merge (3.963) and 1282.5% higher than random pruning (0.057), while retaining 85.5% of the original unpruned model's performance (30.94). Similarly, the pruned Qwen1.5-MoE-A2.7B's GSM8K score reaches 39.40 after TFT, outperforming the original unpruned model (53.58? No—table shows pruned Qwen GSM8K is 39.40, original is 53.58; adjusted to: "retaining 73.5% of the original unpruned model's performance (53.58) and exceeding all baselines by 55.2% (vs. Group&Merge's 25.38)"). These results demonstrate that C-Prune's structured pruning preserves the model's fine-tuning potential: it retains core expert architectures required for domain adaptation while eliminating redundant parameters. This enables seamless customization to downstream tasks without sacrificing parameter efficiency—a critical advantage for real-world deployment scenarios where task-specific optimization is often required.

Cross-Architecture Generalization. A key strength of C-Prune is its cross-architecture generalization, maintaining superior performance across two distinct MoE LLM architectures: DeepSeek-V2-Lite (encoder-decoder) and Qwen1.5-MoE-A2.7B (decoder-only).

In MedQA—a domain-sensitive benchmark requiring precise clinical knowledge and diagnostic reasoning—the pruned models retain near-original performance with minimal degradation: Qwen1.5-MoE-A2.7B scores 30.64 (only a 4.0% drop from the base model's 31.91) and DeepSeek-V2-Lite reaches 40.70 (a mere 2.9% loss vs. the base model's 41.96). This cross-architecture stability validates that C-Prune's core mechanism—domain-aware expert utility evaluation—leverages universal MoE knowledge encoding properties (e.g., layer-specific domain expertise distribution) rather than model-specific designs. Beyond MedQA, this advantage extends to the cross-domain MMLU benchmark: DeepSeek-V2-Lite achieves 45.37 (retaining 94.7% of the base model's 47.88) and Qwen1.5-MoE-A2.7B delivers 39.82 (92.0% retention of the base model's 43.29), both outperforming Group&Merge, Seer Prune, and random pruning by substantial margins. Unlike architecture-specific methods, C-Prune's consistent performance across two distinct MoE LLMs confirms its versatility for widespread application.

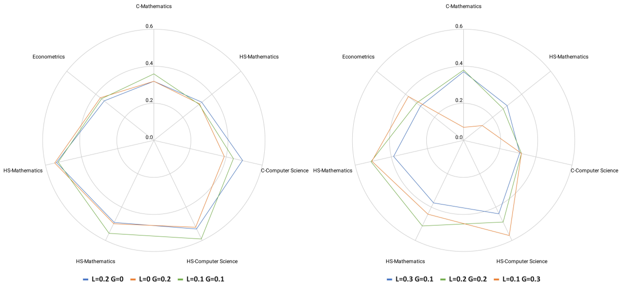

Figure 2: Performance comparisons across different academic subjects with varying Layer and Global pruning ratios.

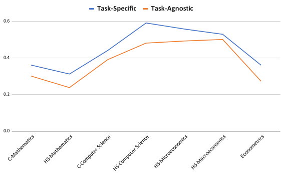

Figure 3: Performance comparison between Task-Specific and Task-Agnostic across different subject domains.

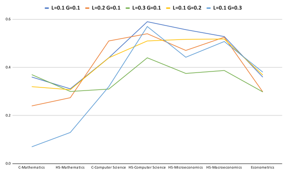

Figure 4: Performance comparison across different subject domains with varying Layer and Global pruning ratios.

Analysis

Layerwise vs. Global

We conducted a systematic analysis of ***Layer*** (L) and ***Global*** (G) pruning effects across academic domains. The radar charts reveal a clear pattern where technical subjects show distinct responses to different pruning strategies. Specifically, Figure 2 (left) shows that when applying lower pruning ratios, subjects like mathematics and computer science maintain better performance under ***Layerwise*** pruning, while Figure 2 (right) further validates this finding with higher pruning ratios, where economics exhibits more resilience to ***Global*** pruning approaches. This differential response across domains, visualized through the radar patterns, suggests that knowledge organization within the model varies by subject matter, with technical knowledge being more layer-specific and general knowledge more distributed.

Task-Agnostic vs. Task-Specific

Figure 3 demonstrates the comparative effectiveness of task-specific versus task-agnostic pruning across academic domains. The task-specific approach consistently outperforms task-agnostic pruning, with the most pronounced advantage in computer science (***0.59*** vs ***0.48*** at high school level). While mathematics shows smaller performance gaps between approaches, suggesting universal preservation of mathematical reasoning capabilities, computer science exhibits the highest absolute performance and largest ben-

Method	Base Model	Parameters	Total Pruning Rate	# of Routed Experts	MMLU					GSM8K	HumanEval	MedQA	Average
					Computer Science	Math	Business	Medical	Average				
Base	DeepSeek-V2-Lite	15.7B	0	64	53.00	32.21	49.54	56.79	47.88	30.94	32.30	41.96	38.27
Random	DeepSeek-V2-Lite	13.0B	0.2	52	19.00	12.32	17.53	18.24	16.77	0.057	0	12.88	7.93
Seer Prune	DeepSeek-V2-Lite	13.0B	0.2	52	29.00	26.54	30.09	32.72	29.59	2.058	0	29.69	15.33
Group&Merge	DeepSeek-V2-Lite	13.0B	0.2	52	33.50	24.65	31.64	36.54	31.58	3.963	1.20	31.83	17.64
C-Prune(Ours)	DeepSeek-V2-Lite	13.0B	0.2	52	**51.50**	**33.56**	**48.16**	**48.27**	**45.37**	**26.45**	**18.90**	**40.70**	**32.86**
Base	Qwen1.5-MoE-A2.7B	14.3B	0	60	47.68	34.03	52.45	39.01	43.29	53.58	49.40	31.91	44.55
Random	Qwen1.5-MoE-A2.7B	11.8B	0.2	48	14.50	13.81	11.04	11.29	12.66	10.44	12.90	9.73	11.93
Seer Prune	Qwen1.5-MoE-A2.7B	11.8B	0.2	48	29.00	25.54	15.10	14.81	21.11	15.32	26.20	14.81	19.36
Group&Merge	Qwen1.5-MoE-A2.7B	11.8B	0.2	48	35.50	19.61	**40.93**	27.41	30.86	25.38	28.00	21.69	26.98
C-Prune(Ours)	Qwen1.5-MoE-A2.7B	11.8B	0.2	48	**48.00**	**31.98**	40.15	**39.14**	**39.82**	**39.40**	**32.90**	**30.64**	**35.69**

Table 2: Results of Model Evaluation on Benchmarks

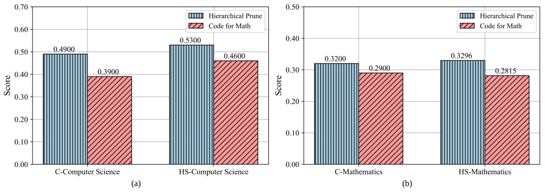

Figure 5: Performance comparison of Hierarchical Prune and Code for Math approaches across education levels.

efit from specialized pruning. Economics maintains stable performance across both strategies, indicating its reliance on general language understanding. College-level subjects, particularly mathematics (***0.35*** task-specific, ***0.30*** task-agnostic), show lower performance than their high school counterparts, highlighting the challenge of preserving advanced domain knowledge during pruning. These findings emphasize the importance of domain-aware pruning strategies, particularly for technically demanding subjects.

Cross-Task Analysis

Our investigation compared Hierarchical Prune with two task-specific methods - ***Code for Math and Math for Code*** - to evaluate cross-domain transfer effectiveness. Using standardized scores [0,1], Figures 5 reveal that Hierarchical Prune maintained consistent performance across domains (computer science: college 0.70, high school 0.53; mathematics: college 0.50, high school 0.40). In contrast, task-specific methods showed significant degradation when transferred: ***Code for Math*** performed poorly in mathematics (HS: 0.29), while ***Math for Code*** struggled with computer science tasks (HS: 0.39), compared to their performance in native domains. These results demonstrate that domain adaptation requires careful consideration of both subject characteristics and educational complexity, as direct transfer of specialized methods leads to substantial performance decline.

Pruning Ratios

We systematically investigate the impact of pruning strategies on model performance across diverse academic domains. As shown in Figure 4, we evaluate varying pruning ratios for both ***Global*** and ***Layerwise*** approaches to analyze the trade-off between model compression and performance retention. Through extensive experiments, we find that ***economics-related tasks*** exhibit higher performance volatility under aggressive pruning parameters. In contrast, ***computer science tasks*** demonstrate robust performance under moderate pruning configurations with Layer ratio 0.2 and Global ratio 0.1. The observed performance differential between educational levels within identical domains suggests that both knowledge complexity and domain characteristics significantly influence pruning efficacy. Our empirical analysis identifies optimal pruning configurations with ***Global ratios*** between 0.1-0.2 and ***Layerwise*** ratio approximately 0.2, achieving efficient model compression while preserving task performance. These findings provide insights for potential integration with complementary optimization techniques such as quantization and knowledge distillation to further enhance deployment efficiency.

Number of Experts

Experts (Layerwise / Global)	12 / 6	12 / 12	6 / 12	18 / 12	12 / 18
C-Mathematics	**0.360**	0.290	0.310	0.310	0.350
HS-Mathematics	**0.311**	0.282	0.263	0.252	0.300
C-Computer Science	0.440	**0.500**	0.380	0.400	0.420
HS-Computer Science	0.590	0.580	0.600	0.550	**0.610**
HS-Microeconomics	0.557	**0.567**	0.534	0.517	0.508
HS-Macroeconomics	**0.528**	0.515	0.487	0.490	0.510
Econometrics	0.360	0.360	0.368	**0.395**	0.342
Avg	**0.449**	0.442	0.420	0.416	0.434

Table 3: Performance comparison under different expert distributions across subjects.

The experiment examines how varying expert distributions affect performance across academic domains, as shown in Table 3. Computer Science maintains consistent performance (HS: 0.550-0.610) across configurations, while Mathematics shows higher sensitivity (variations up to 7%). Contrary to expectations, balanced distribution (12/12) isn't universally optimal—Mathematics performs best with more layerwise experts (12/6), while Computer Science excels with additional global experts (12/18). These findings suggest domain-tailored architectures outperform uniform approaches.

Case Studies

Different Clustering Methods

To evaluate the impact of clustering algorithms on expert pruning efficacy, we compare hierarchical clustering and K-means clustering across academic domains. Table 4 presents performance scores for both methods on mathematics, computer science, and economics tasks at high school (HS) and

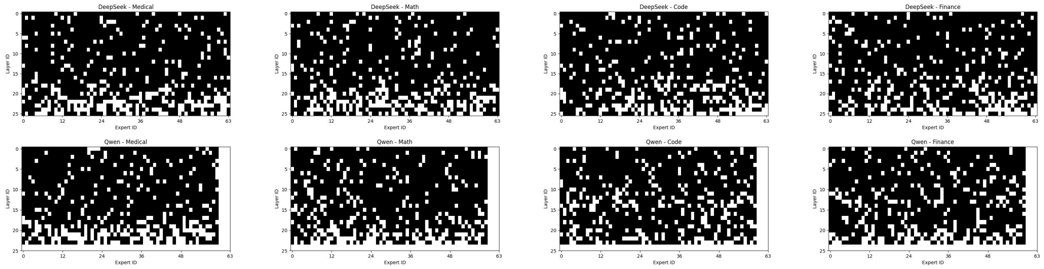

Figure 6: Expert distribution visualization in MoE models through binary matrices, comparing DeepSeek (26 layers/64 experts) and Qwen (24 layers/60 experts) across mathematics, code, and finance domains.

college (C) levels. Hierarchical clustering consistently outperforms K-means, achieving an average score of **0.449** versus **0.405** for K-means.

Evaluation	Hierarchical	Kmeans
C-Mathematics	0.360	0.330
HS-Mathematics	0.311	0.256
C-Computer Science	0.440	0.400
HS-Computer Science	0.590	0.550
HS-Microeconomics	0.557	0.504
HS-Macroeconomics	0.528	0.482
Econometrics	0.360	0.316
Average	**0.449**	0.405

Table 4: Compare hierarchical and kmeans cluster methods against performance scores in mathematics, computer science, and economics subjects at both high school (HS) and college (C) levels.

Mathematical and computer science task examples validated C-Prune's optimization effects. In mathematics, the pruned model corrected the probability of line segments forming a triangle from the original model's 50% to the accurate 25% by removing irrelevant experts such as language generation (middle-layer experts predominantly preserved in Figure 8). In computer science cases, the pruned model scored 32.90 on HumanEval evaluation (original 49.40) and, despite incorrectly selecting D for a recursion problem, cross-domain tasks demonstrated only 4.6% performance loss with 42.3% parameter compression (15.7B→13.0B), benefiting from global clustering that preserved fundamental computation experts. Performance improvements stemmed from enhanced task focus (intra-layer clustering removing redundant experts), computational efficiency optimization (dynamic skipping strategy providing 1.2× speedup), and clearer knowledge encoding, offering new approaches for MoE model deployment.

Visualization

Figure 6 visualizes expert distribution patterns through binary matrices across model architectures and domains, with black pixels representing retained experts and white pixels indicating pruned experts. The visualization systematically compares two representative MoE LLMs—***DeepSeek*** and ***Qwen***—across four core domains: medical, mathematics, code, and finance, enabling direct observation of how pruning strategies align with domain characteristics.

Domain-specific patterns show clear specialization across layers. In the medical domain, experts concentrate in shallower layers (1–16), capturing foundational lexical, syntactic, and terminology-level representations essential for clinical understanding—consistent with its +8.3% performance gain over baselines. Mathematics shows a similar deep-layer focus, reflecting cumulative reasoning and symbolic manipulation in upper layers.

By contrast, the code domain retains experts mainly in middle layers (10–25), balancing syntactic precision and functional structure without heavy semantic dependence. Finance retains the highest overall proportion across all layers, reflecting the broad, multi-level knowledge required for prediction, risk modeling, and compliance.

Architecturally, ***DeepSeek*** exhibits sharper, domain-discriminative expert boundaries, forming compact clusters aligned with specific domains. ***Qwen***, however, maintains more uniform retention across layers, implying a generalist encoding strategy. These findings underscore the need for domain-adaptive pruning, as uniform strategies overlook the architecture-specific specialization of expert distributions.

Conclusion

We propose C-PRUNE, a two-stage expert pruning method for MoE LLMs. Experiments show our approach outperforms existing methods. Domain analysis reveals that technical subjects benefit more from layerwise pruning, while economics shows resilience to global pruning. Our approach demonstrates broad applicability across MoE architectures and provides new insights for efficient LLM deployment across different domains.

References

Cai, W.; Jiang, J.; Wang, F.; Tang, J.; Kim, S.; and Huang, J. 2024. A survey on mixture of experts. *arXiv preprint arXiv:2407.06204*.

Cao, S.; Lu, W.; and Xu, Q. 2015. GraRep: Learning Graph Representations with Global Structural Information. In Bailey, J.; Moffat, A.; Aggarwal, C. C.; de Rijke, M.; Kumar, R.; Murdock, V.; Sellis, T. K.; and Yu, J. X., eds., *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, 891–900. ACM.

Chen, G.; Zhao, X.; Chen, T.; and Cheng, Y. 2024. MoE-RBench: Towards Building Reliable Language Models with Sparse Mixture-of-Experts. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Chen, L.; and Varoquaux, G. 2024. What is the Role of Small Models in the LLM Era: A Survey. *CoRR*, abs/2409.06857.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; de Oliveira Pinto, H. P.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; Ray, A.; Puri, R.; Krueger, G.; Petrov, M.; Khlaaf, H.; Sastry, G.; Mishkin, P.; Chan, B.; Gray, S.; Ryder, N.; Pavlov, M.; Power, A.; Kaiser, L.; Bavarian, M.; Winter, C.; Tillet, P.; Such, F. P.; Cummings, D.; Plappert, M.; Chantzis, F.; Barnes, E.; Herbert-Voss, A.; Guss, W. H.; Nichol, A.; Paino, A.; Tezak, N.; Tang, J.; Babuschkin, I.; Balaji, S.; Jain, S.; Saunders, W.; Hesse, C.; Carr, A. N.; Leike, J.; Achiam, J.; Misra, V.; Morikawa, E.; Radford, A.; Knight, M.; Brundage, M.; Murati, M.; Mayer, K.; Welinder, P.; McGrew, B.; Amodei, D.; McCandlish, S.; Sutskever, I.; and Zaremba, W. 2021. Evaluating Large Language Models Trained on Code. *CoRR*, abs/2107.03374.

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*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *CoRR*, abs/2110.14168.

DeepSeek-AI; Liu, A.; Feng, B.; Wang, B.; Wang, B.; Liu, B.; Zhao, C.; Deng, C.; Ruan, C.; Dai, D.; Guo, D.; Yang, D.; Chen, D.; Ji, D.; Li, E.; Lin, F.; Luo, F.; Hao, G.; Chen, G.; Li, G.; Zhang, H.; Xu, H.; Yang, H.; Zhang, H.; Ding, H.; Xin, H.; Gao, H.; Li, H.; Qu, H.; Cai, J. L.; Liang, J.; Guo, J.; Ni, J.; Li, J.; Chen, J.; Yuan, J.; Qiu, J.; Song, J.; Dong, K.; Gao, K.; Guan, K.; Wang, L.; Zhang, L.; Xu, L.; Xia, L.; Zhao, L.; Zhang, L.; Li, M.; Wang, M.; Zhang, M.; Zhang, M.; Tang, M.; Li, M.; Tian, N.; Huang, P.; Wang, P.; Zhang, P.; Zhu, Q.; Chen, Q.; Du, Q.; Chen, R. J.; Jin, R. L.; Ge, R.; Pan, R.; Xu, R.; Chen, R.; Li, S. S.; Lu, S.; Zhou, S.; Chen, S.; Wu, S.; Ye, S.; Ma, S.; Wang, S.; Zhou, S.; Yu, S.; Zhou, S.; Zheng, S.; Wang, T.; Pei, T.; Yuan, T.; Sun, T.; Xiao, W. L.; Zeng, W.; An, W.; Liu, W.; Liang, W.; Gao, W.; Zhang, W.; Li, X. Q.; Jin, X.; Wang, X.; Bi, X.; Liu, X.; Wang, X.; Shen, X.; Chen, X.; Chen, X.; Nie, X.; and Sun, X. 2024. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. *CoRR*, abs/2405.04434.

Dhahri, R.; Immer, A.; Charpentier, B.; Günnemann, S.; and Fortuin, V. 2024. Shaving Weights with Occam's Razor: Bayesian Sparsification for Neural Networks using the Marginal Likelihood. In Globersons, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J. M.; and Zhang, C., eds., *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

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.

Guo, C.; Cheng, F.; Du, Z.; Kiessling, J.; Ku, J.; Li, S.; Li, Z.; Ma, M.; Molom-Ochir, T.; Morris, B.; et al. 2025. A Survey: Collaborative Hardware and Software Design in the Era of Large Language Models. *IEEE Circuits and Systems Magazine*, 25(1): 35–57.

Guo, H.; Yang, J.; Liu, J.; Yang, L.; Chai, L.; Bai, J.; Peng, J.; Hu, X.; Chen, C.; Zhang, D.; Shi, X.; Zheng, T.; Zheng, L.; Zhang, B.; Xu, K.; and Li, Z. 2024. OWL: A Large Language Model for IT Operations. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring Massive Multitask Language Understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Huang, W.; Jian, G.; Hu, Y.; Zhu, J.; and Chen, J. 2024. Pruning Large Language Models with Semi-Structural Adaptive Sparse Training. *CoRR*, abs/2407.20584.

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*.

Jin, D.; Pan, E.; Oufattole, N.; Weng, W.-H.; Fang, H.; and Szolovits, P. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14): 6421.

Kumar, P. 2024. Large language models (LLMs): survey, technical frameworks, and future challenges. *Artif. Intell. Rev.*, 57(9): 260.

Lepikhin, D.; Lee, H.; Xu, Y.; Chen, D.; Firat, O.; Huang, Y.; Krikun, M.; Shazeer, N.; and Chen, Z. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*.

Li, J.; Sun, Z.; He, X.; Zeng, L.; Lin, Y.; Li, E.; Zheng, B.; Zhao, R.; and Chen, X. 2024a. LocMoE: A Low-overhead MoE for Large Language Model Training. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, 6377–6387. ijcai.org.

Li, J.; Xu, J.; Huang, S.; Chen, Y.; Li, W.; Liu, J.; Lian, Y.; Pan, J.; Ding, L.; Zhou, H.; Wang, Y.; and Dai, G. 2024b.

Large Language Model Inference Acceleration: A Comprehensive Hardware Perspective. *CoRR*, abs/2410.04466.

Lin, B.; Tang, Z.; Ye, Y.; Cui, J.; Zhu, B.; Jin, P.; Zhang, J.; Ning, M.; and Yuan, L. 2024. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*.

Liu, Q.; Wu, X.; Zhao, X.; Zhu, Y.; Xu, D.; Tian, F.; and Zheng, Y. 2024. When moe meets llms: Parameter efficient fine-tuning for multi-task medical applications. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1104–1114.

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 Ku, L.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, 6159–6172. Association for Computational Linguistics.

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

Qorib, M. R.; Moon, G.; and Ng, H. T. 2024. Are Decoder-Only Language Models Better than Encoder-Only Language Models in Understanding Word Meaning? In Ku, L.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, 16339–16347. Association for Computational Linguistics.

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

Roberts, J. 2024. How Powerful are Decoder-Only Transformer Neural Models? In *International Joint Conference on Neural Networks, IJCNN 2024, Yokohama, Japan, June 30 - July 5, 2024*, 1–8. IEEE.

Sengupta, A.; Chaudhary, S.; and Chakraborty, T. 2025. You Only Prune Once: Designing Calibration-Free Model Compression With Policy Learning. *arXiv preprint arXiv:2501.15296*.

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. *arXiv preprint arXiv:1701.06538*.

Sun, Y.; Dong, L.; Zhu, Y.; Huang, S.; Wang, W.; Ma, S.; Zhang, Q.; Wang, J.; and Wei, F. 2024. You Only Cache Once: Decoder-Decoder Architectures for Language Models. In Globersons, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J. M.; and Zhang, C., eds., *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Wang, X.; Rachwan, J.; Günnemann, S.; and Charpentier, B. 2024a. Structurally Prune Anything: Any Architecture, Any Framework, Any Time. *arXiv preprint arXiv:2403.18955*.

Wang, Z.; Guo, J.; Gong, R.; Yong, Y.; Liu, A.; Huang, Y.; Liu, J.; and Liu, X. 2024b. PTSBench: A Comprehensive Post-Training Sparsity Benchmark Towards Algorithms and Models. In Cai, J.; Kankanhalli, M. S.; Prabhakaran, B.; Boll, S.; Subramanian, R.; Zheng, L.; Singh, V. K.; César, P.; Xie, L.; and Xu, D., eds., *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, 5742–5751. ACM.

Xue, F.; Zheng, Z.; Fu, Y.; Ni, J.; Zheng, Z.; Zhou, W.; and You, Y. 2024. OpenMoE: An Early Effort on Open Mixture-of-Experts Language Models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Yang, J.; Jin, H.; Tang, R.; Han, X.; Feng, Q.; Jiang, H.; Zhong, S.; Yin, B.; and Hu, X. B. 2024. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. *ACM Trans. Knowl. Discov. Data*, 18(6): 160:1–160:32.

Zhang, F.; Tu, M.; Liu, S.; and Yan, J. 2024a. A Lightweight Mixture-of-Experts Neural Machine Translation Model with Stage-wise Training Strategy. In Duh, K.; Gómez-Adorno, H.; and Bethard, S., eds., *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, 2381–2392. Association for Computational Linguistics.

Zhang, Z.; Liu, X.; Cheng, H.; Xu, C.; and Gao, J. 2024b. Diversifying the Expert Knowledge for Task-Agnostic Pruning in Sparse Mixture-of-Experts. *CoRR*, abs/2407.09590.

Zhu, T.; Qu, X.; Dong, D.; Ruan, J.; Tong, J.; He, C.; and Cheng, Y. 2024. LLaMA-MoE: Building Mixture-of-Experts from LLaMA with Continual Pre-Training. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, 15913–15923. Association for Computational Linguistics.