Unveiling Hidden Collaboration within Mixture-of-Experts in Large Language Models

Anonymous ACL submission

Abstract

Mixture-of-Experts based large language models (MoE LLMs) have shown significant promise in multitask adaptability by dynamically routing inputs to specialized experts. Despite their success, the collaborative mechanisms among experts are still not well understood, limiting both the interpretability and optimization of these models. In this paper, we focus on two critical issues: (1) identifying expert collaboration patterns, and (2) optimizing MoE LLMs through expert pruning. To address the first issue, we propose a hierarchical sparse dictionary learning (HSDL) method that uncovers the collaboration patterns among experts. For the second issue, we introduce the Contribution-Aware Expert Pruning (CAEP) algorithm, which effectively prunes low-contribution experts. Our extensive experiments demonstrate that expert collaboration patterns are closely linked to specific input types and exhibit semantic significance across various tasks. Moreover, pruning experiments show that our approach improves overall performance by 2.5% on average, outperforming existing methods. These findings offer valuable insights into enhancing the efficiency and interpretability of MoE LLMs, offering a clearer understanding of expert interactions and improving model optimization. The code repository is available at this URL.

1 Introduction

In recent years, the MoE LLMs have gained significant attention as a computationally efficient framework, demonstrating exceptional representational power for large-scale machine learning tasks (Jiang et al., 2024; Fedus et al., 2022). By leveraging a dynamic routing mechanism, MoE enables the collaborative operation of specialized "Experts", each designed to process complex input data. Compared to traditional architectures, MoE LLMs offer more flexible and adaptive knowledge representations while reducing computational costs, making them well-suited for resource-intensive situations (Cai et al., 2024).

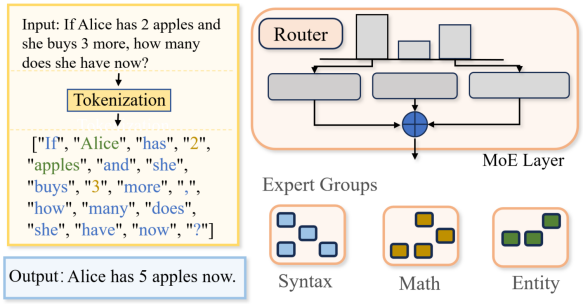

Figure 1: In MoE LLMs, a group of experts often collaborate to analyze a certain type of tokens, and they are not necessarily in the same layer.

Existing research on understanding the working mechanism of MoE LLMs has largely focused on analyzing the behavior of the router, which governs expert selection (Lo et al., 2024). For instance, some studies highlight the influence of output norms on expert selection (Lo et al., 2024), while others reveal that token IDs play a significant role in routing decisions (Jiang et al., 2024; Xue et al., 2024; Dai et al., 2024). These efforts have provided valuable insights into how MoE LLMs allocate tasks to specialized experts, enhancing multitask adaptability.

Despite the widespread success of MoE LLMs, several key challenges remain underexplored. One of the main challenges is understanding the collaborative mechanisms among the experts within the network. While MoE LLMs generate final outputs by combining the predictions of multiple experts, how these experts cooperate to produce the outputs is still not well understood. Figure 1 conceptualizes the notion of cross-layer expert collaboration - coordinated groups of experts across distinct layers that exhibit synchronized activation to implement specific functional modules. This phenomenon is empirically validated in operational MoE networks.

Figure 2 illustrates a representative case of strong co-activation patterns between Expert 21 in Layer 5 and Expert 3 in Layer 6. Comprehending these collaboration patterns is essential, as it directly influences knowledge sharing, model interpretability, performance, and optimization. Another key challenge lies in the high model complexity of MoE LLMs, which presents significant challenges in terms of deployment, limiting their scalability for large-scale applications (Lu et al., 2024; He et al., 2024).

Therefore, this study aims to investigate and reveal the collaboration patterns between experts in MoE LLMs, and utilize these patterns to enhance model efficiency and performance. The core questions we address include: (1) Are there consistent collaboration patterns among experts, and what do they reveal about the tasks implicitly learned in MOE LLMs? (2) Can these collaboration patterns be leveraged to compress MoE LLMs?

To address the two key questions, we begin by extracting the expert activation matrix, which serves as the foundation for further analysis. For the first question, we apply a novel hierarchical sparse dictionary learning (HSDL) approach to uncover collaboration structures within the expert activation data. Building on these insights, we then investigate expert pruning through the Contribution-Aware Expert Pruning (CAEP) algorithm, which identifies and removes low-contribution experts. This process reduces model redundancy, alleviating storage pressure while preserving or even enhancing performance. The entire pipeline, as outlined in Figure 3, comprises three key components: (1) Expert Activation Data Collection, (2) MoE Collaboration Pattern Mining, and (3) Expert Pruning Based on Expert Collaboration Pattern.

In our experimental evaluation, we tested several representative MoE architectures, including the DeepSeek model, on the MMLU-pro dataset, which contains 2,812 samples across five chosen domains: mathematics, computer science, physics, law, and psychology. Our analysis of the learned dictionaries revealed domain-specific expert collaboration patterns with distinct semantic significance. Building on these insights, we conducted pruning experiments using the CAEP method, which demonstrated that pruning experts based on these patterns effectively reduces the number of experts while maintaining or even improving performance. Our method outperforms baselines with an average improvement of 2.5%, and in the best case, pruning

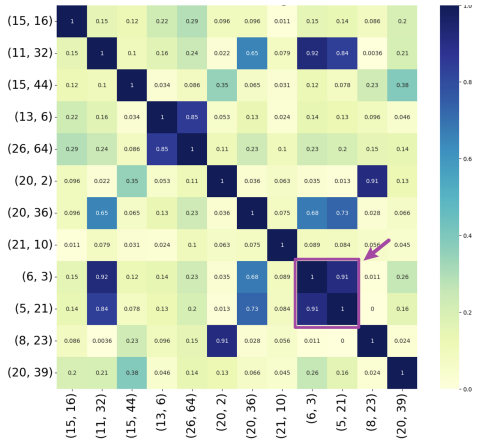

Figure 2: Here (x, y) refers to the y-th expert in x-th layer. By selecting any two experts from the MoE, we can calculate the probability of their co-activation. It can be observed that Expert 21 from the layer 5 and Expert 3 from the layer6 frequently activate simultaneously, forming an expert collaboration pattern.

50% of experts results in only a 5.7% performance drop for specific tasks.

Our contribution can be summarized as follows:

- We explore and uncover the latent collaboration patterns among experts in MoE LLMs. We propose hierarchical sparse dictionary learning (HSDL) and reveal how experts interact and cooperate, which provides new insights into the collaborative mechanisms that drive the performance of MoE LLMs.
- We propose the Contribution-Aware Expert Pruning (CAEP) algorithm, which optimizes model efficiency by pruning low-contribution experts without sacrificing performance. Our experiments show that CAEP maintains competitive performance while significantly reducing the number of experts, effectively balancing pruning and performance retention.

2 Literature Review

2.1 Analysis of Routing in MoE Networks

The analysis of router behavior in MoE networks focuses on understanding how the model selects experts based on input features, which is key for optimizing performance. For instance, Lo et al. found that routers typically select experts with larger output norms (Lo et al., 2024), while other studies suggest that router choices are more related to token IDs than to expert fields (Jiang et al., 2024; Xue et al., 2024; Dai et al., 2024). While these approaches offer valuable insights, they often treat

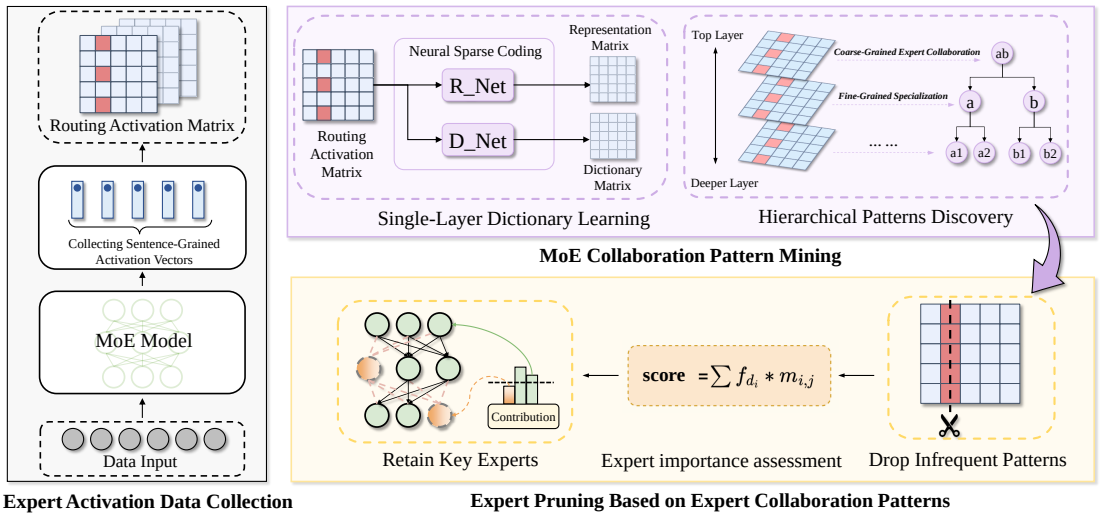

Figure 3: Overview of Our Study's Pipeline.

experts as independent entities, overlooking the collaboration patterns between them.

2.2 Expert Pruning in MoE

Expert pruning reduces storage consumption in MoE networks by removing less impactful experts. Current strategies include: (1) discarding experts with low activation frequencies based on router decisions (Muzio et al., 2024), (2) identifying experts with minimal output influence using $|x - f(x)|$ differences (Lu et al., 2024; He et al., 2024), and (3) merging experts by calculating weight similarities (Li et al., 2023; Zhang et al., 2024). However, these methods often treat experts independently or focus on merging similar groups, without exploring diverse expert combinations with distinct roles.

2.3 Sparse Dictionary Learning

Sparse dictionary learning is a well-established method in representation learning and dimensionality reduction (Yang et al., 2010; Wright et al., 2009). It constructs a dictionary of features that enables sparse representation of data, facilitating efficient encoding of high-dimensional information (Tang et al., 2023; Chen et al., 2013). This approach has proven effective in various applications, such as image processing and signal recovery, where it helps capture essential features while reducing noise (Hou et al., 2021, 2020). Recently, companies like OpenAI, Google, and Anthropic have applied sparse dictionary learning to understand large language models' mechanisms (Rajamanoharan et al., 2024; Gao et al.). Despite its success in other areas, sparse dictionary learning has been underutilized in explanatory research on MoE networks.

3 Extraction of Expert Activation Matrix

In MoE LLMs, the activation weights of the experts reflect the intensity of their responses to the input data, thereby elucidating the collaborative patterns among them. Furthermore, these activation data provide a foundational basis for optimizing pruning strategies, which in turn contribute to enhanced computational and storage efficiency. Consequently, the extraction and analysis of activation weights are critical steps in the effective exploration of collaboration patterns and the implementation of pruning techniques.

Given an MoE LLM with m layers and n experts, and an input dataset S containing N_s samples, we extract the expert activation data to construct a two-dimensional activation tensor $V \in \mathbb{R}^{N_s \times (m \times n)}$, where each element $v_{i,j,k}$ represents the activation weight of the k-th expert in the j-th layer for the i-th sample. This activation weight quantifies the intensity of the expert's response to the input sample, with values constrained within the range $[0, 1]$.

To aggregate the activation data of each sample into a sentence-level representation, we sum the activation values of all tokens within a sample, thereby obtaining the sentence-level activation value for each layer. Let $\alpha(i)_{t,j,k}$ denote the routing allocation of the t-th token in sample S_i to the k-th expert in the j-th layer. The sentence-level activation value is then computed as:

$$v_{i,j,k} = \sum_{t=1}^{T} \alpha(i)_{t,j,k}, \quad (1)$$

where T represents the sequence length. Finally, by transposing and accumulating these activation data, we construct the expert activation matrix X, which serves as the input to the subsequent analysis of collaboration patterns among experts.

4 MoE Collaboration Pattern Mining

In this section, we propose a novel **Hierarchical Sparse Dictionary Learning (HSDL)** approach to uncover collaboration patterns among experts in MoE LLMs through hierarchical decomposition. Furthermore, We evaluate its effectiveness on the MMLU-pro dataset, validating the method by comparing it to exhaustive search techniques and exploring domain-specific expert interactions, demonstrating its versatility and efficiency in capturing complex MoE dynamics.

4.1 Problem Definition

The objective of this task is to extract the collaboration patterns among experts in MoE LLMs. Given a dataset $S = \{s_1, s_2, \ldots, s_{N_s}\}$ comprising N_s samples, we construct an expert activation matrix $X \in \mathbb{R}^{N_e \times N_s}$, where N_e denotes the total number of experts. By employing sparse dictionary learning techniques to decompose X, we obtain a dictionary matrix $D \in \mathbb{R}^{N_e \times N_p}$ and a sparse coding matrix $R \in \mathbb{R}^{N_p \times N_s}$, with N_p representing the predefined dictionary capacity. Our goal is to decompose the expert activation matrix $\mathbf{X}$ into a dictionary matrix $\mathbf{D}$ and a sparse coding matrix $\mathbf{R}$, which can be expressed as $\mathbf{X} \approx \mathbf{D} \cdot \mathbf{R}$.

Here, the dictionary matrix D encodes the collaboration patterns among experts, while the sparse coding matrix R determines how these patterns combine to reconstruct X.

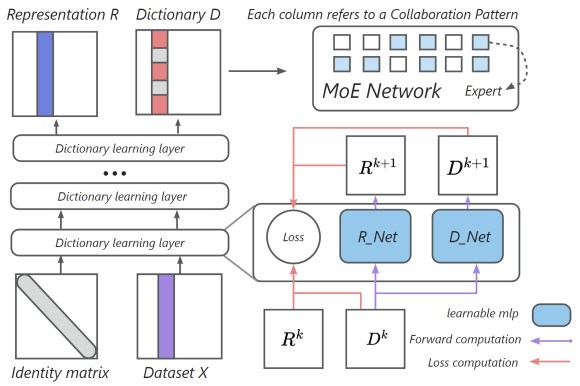

Figure 4: Hierarchical Sparse Dictionary Learning.

4.2 Hierarchical Sparse Dictionary Learning for Expert Collaboration Patterns Mining

Sparse dictionary learning is an effective unsupervised method for uncovering latent structures in data through sparse representations. By modeling data as a linear combination of dictionary atoms, it reveals expert collaboration patterns in MoE LLMs. However, a single-layer approach fails to capture complex patterns across varying granularities. To address this, we propose the HSDL approach, which recursively decomposes the dictionary matrix, capturing collaboration patterns from coarse to fine granularity, thus revealing multi-layered expert interactions.

We extend the original single-layer structure decomposition into a hierarchical structure by recursively decomposing the dictionary matrix at each layer k into finer subpatterns represented by D_{k+1}, formulated as $D_k \approx D_{k+1} \cdot R_{k+1}$.

Figure 4 illustrates the hierarchical structure of Sparse Dictionary Learning, showing how the multi-layered expert collaboration is modeled across different layers.

Furthermore, we introduce three key constraints to optimize the multi-layer dictionary learning process:

(1) **Dictionary Size Constraint**: This loss term is designed to limit the dictionary size in order to obtain a more compact dictionary, preventing certain dictionary elements from dominating. Specifically, $R_{k,i,:}$ denotes the sparse coding of the i-th data point at layer k. This constraint is defined as:

$$L_{\text{sparse}} = \sum_{k=0}^{K} \sum_{i=0}^{M} \|R_{k,i,:}\|_{\infty}. \tag{2}$$

(2) **Sparsity Constraint**: This ensures that the sparse coding matrix R_k at each layer remains sparse. The matrix $R_{k,:,j}$ represents the contribution of the j-th dictionary atom at layer k. The formula is:

$$L_{\text{hier}} = \sum_{k=0}^{K} \sum_{j=0}^{M} \|R_{k+1,:,j}\|_1 \cdot \|R_{k,:,j}\|_1 / N. \tag{3}$$

(3) **Reconstruction Error Term**: This ensures that the relationships between dictionaries at successive layers are consistently learned. The reconstruction error is defined as:

$$L_{\text{rec}} = \sum_{k=0}^{K} \sum_{j=0}^{M} \|D_{k,:,j} - (D_{k+1} R_{k+1})_{:,j}\|_1 \cdot \frac{\|R_{k,:,j}\|_1}{N}. \tag{4}$$

These three constraints collectively guide the optimization of both the hierarchical dictionary and sparse coding matrices. The overall loss function is formulated as:

$$L_{\text{total}} = L_{\text{sparse}} + \lambda_1 L_{\text{hier}} + \lambda_2 L_{\text{rec}}, \quad (5)$$

where λ_1 and λ_2 are hyperparameters that control the respective losses. By minimizing this loss function, we optimize both the dictionary matrix D_k and the sparse coding matrix R_k at each layer, effectively capturing the multi-level structure of expert collaboration.

4.3 Experimental Analysis of Expert Collaboration Patterns

In this subsection, we aim to explore how the collaboration patterns among experts in MoE-based LLMs reflect the tasks implicitly learned by the model, thereby contributing to a deeper understanding of its functioning. We present a detailed analysis of the expert collaboration patterns identified through our hierarchical sparse dictionary learning method.

4.3.1 Experimental Setup

We use the Phi-MoE model and apply our HSDL method to 2,812 samples from the MMLU-pro dataset, covering five domains: mathematics, computer science, physics, law, and psychology.

4.3.2 Prompt Interpretation using Expert Collaboration Pattern

To explore how expert collaboration patterns in MoE LLMs reflect the model's understanding of tasks, we conduct a detailed analysis using the hierarchical dictionary learning method. Specifically, we aim to understand how different experts collaborate to handle specific aspects of a problem.

To achieve this, we designed a semantic annotation scheme for input sentences to interpret the semantics of expert collaboration patterns derived from HSDL. We color words processed by the same dictionary atoms (i.e., expert collaboration patterns) with the same color. This color-coding scheme facilitates the observation of interrelationships of the expert collaboration patterns. We analyze the input samples using the dictionary atoms obtained through HSDL, with one such analysis shown in Figure 5.

Results and Discussion. We find that the hierarchical semantic annotation of expert collaboration patterns reveals how MoE LLMs understand and process different tasks within a problem.

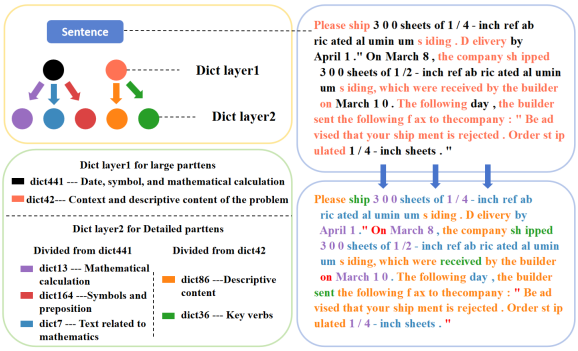

Figure 5: Hierarchical Semantic Annotation of Dictionary Elements on MMLU.

As shown in Figure 5, in the upper left corner, we can observe that: **Expert collaboration patterns in higher-layer and lower-layer dictionaries demonstrate a hierarchical semantic relationship, which becomes increasingly fine-grained as layer increases.** The lower left corner of the figure displays this from a semantic perspective, where the top layer captures broad categories such as "Date, symbol, and mathematical calculation," while deeper layers break these down into more detailed components like "Mathematical calculation" or "Key verbs" (See Appendix E for more examples).

These findings provide a direct answer to our central question on expert collaboration patterns in MoE LLMs. The hierarchical decomposition offers a more detailed understanding of the model's internal processes, shedding light on how tasks are learned and executed. This approach could evolve into a tool for visualizing MoE LLMs behavior, enhancing interpretability and supporting optimization for domain-specific applications.

4.3.3 Comparison with Exhaustive Search Results

To investigate whether the top dictionary elements correspond to the most frequent expert combinations, we compared the dictionary's expert collaboration patterns with those from an exhaustive search method. Due to the high computational cost of considering larger combinations, we limited this analysis to expert collaboration patterns formed by only two or three experts.

To quantify how well our dictionary captures the most frequent expert combinations, we define N_{top} as the number of dictionary items in the top $k\%$ of the traversal pattern, and N_{total} as the total number of dictionary items. The coverage is then

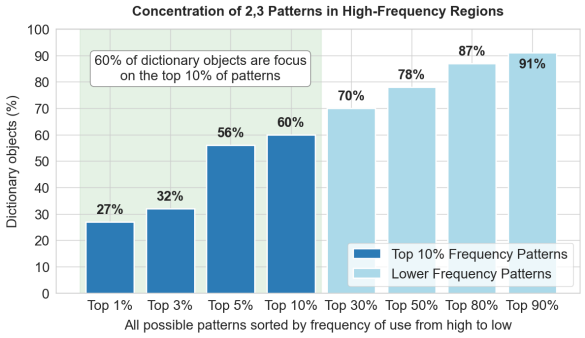

Figure 6: Comparison of overlap with the results of the exhaustive method.

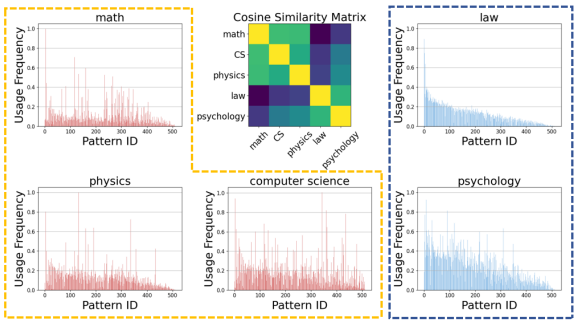

Figure 7: The distribution of expert selection frequencies during inputs from different fields.

calculated using the following formula:

$$\text{Top } k\% \text{ Coverage} = \frac{N_{\text{top}}}{N_{\text{total}}}. \tag{6}$$

Results and Discussion. As shown in Figure 6, **the collaboration patterns identified by our method predominantly align with the most frequent expert combinations found during the exhaustive search.** Specifically, 60% of the patterns identified by our method correspond to the top 10% of the most frequent expert combinations, indicating that our method efficiently identifies the most prevalent collaboration patterns.

While our method focuses on the most frequent expert combinations, it also captures some low-frequency patterns. These less frequent combinations, though less common, are critical for capturing the diversity of expert interactions, which enhances the model's ability to tackle a wider range of tasks. **This highlights the importance of considering both high and low-frequency expert combinations in shaping the performance and versatility of MoE LLMs.**

4.3.4 Domain-Specific Expert Collaboration Patterns

In this experiment, our goal is to explore how expert collaboration patterns vary across different domains and to understand the domain-specific nature of expert interactions within MoE LLMs. Specifically, we aim to examine the activation frequencies of experts for inputs from various fields, including mathematics, computer science, physics, law, and psychology, to uncover potential domain-related patterns.

we analyzed the frequency distribution of activated experts during the model processing for inputs from different domains and calculated the cosine similarity between the distributions of each domain, resulting in a confusion matrix.

Results and Discussion. Figure 7 shows the expert selection frequency distribution across domains. We can observe that for inputs from different fields, the distribution of expert activation frequencies in the MoE LLM varies. For semantically similar domains, such as mathematics, physics, and computer science indicated by the orange dashed box, their distributions are closer to each other. In contrast, the distributions of expert activation frequencies are more different for domains with greater semantic differences, such as mathematics and law. **This suggests that expert collaboration is more specialized within specific domains, reflecting domain-specific interactions in MoE LLMs.**

These findings indicate that **experts in MoE LLMs exhibit domain preferences, adjusting expert selection based on the input domain's characteristics to optimize performance for domain-specific tasks.** Understanding these patterns can enhance the model's efficiency and its ability to handle specialized tasks.

5 Expert Pruning Based on Expert Collaboration Patterns

In this section, we present the CAEP method, which utilizes expert collaboration patterns to reduce the number of experts in an MoE LLM while preserving performance. We first introduce the pruning algorithm and then demonstrate its effectiveness through two types of experiments: (1) General Tasks Evaluation, where we compare CAEP with baseline methods on diverse tasks, and (2) Domain-Specific Evaluation, where we assess its ability to retain domain-relevant capabilities after pruning.

5.1 Pruning algorithm

We propose the **Contribution-Aware Expert Pruning (CAEP)** algorithm. The algorithm aims to produce a mask vector that incorporates our retention strategy, given a specific pruning ratio k. This pruning process is achieved by progressively discarding less significant dictionary atoms, guided by the contribution scores derived from R. The CAEP algorithm proceeds as follows (Algorithm 1):

- **Calculation and Ranking**: Calculate the contribution scores for each expert by the sparse representation matrix R and the dictionary matrix D, obtaining the total contribution and sorting it in descending order.
- **Initial Threshold Mask**: Determine the score based on the predefined threshold ratio and generate the initial binary mask, marking the experts whose contribution scores are above.
- **Iterative Pruning**: Before reaching the target pruning ratio, repeatedly identify the least used patterns and remove them from the dictionary and the sparse representation while updating the contribution scores and the mask, until only the desired ratio of experts remains.

Algorithm 1 Expert Pruning Strategy

Require: Dictionary matrix $\mathbf{D} \in \mathbb{R}^{N_e \times N_p}$
Sparse representation matrix $\mathbf{R} \in \mathbb{R}^{N_p \times N_s}$
Threshold ratio $k_1 \in (0, 1)$
Target pruning ratio $k_2 \in (0, 1)$
Ensure: Pruned expert mask $\mathbf{m} \in \{0, 1\}^{N_e}$
$\mathbf{R}_{\text{sum}} \leftarrow \sum_{j=1}^{N_s} \mathbf{R}_{:,j}$ ▷ Sum over samples
$\mathbf{D}_{\text{sum}} \leftarrow \mathbf{D} \cdot \mathbf{R}_{\text{sum}}^{\top}$ ▷ Weighted by pattern frequency
$\mathbf{e} \leftarrow \sum_{i=1}^{N_p} \mathbf{D}_{\text{sum},i}$ ▷ Aggregate expert contributions
Sort $\mathbf{e}$ in descending order: $\mathbf{e}_{\text{sorted}}$
$f \leftarrow \mathbf{e}_{\text{sorted}}[\lceil k_1 \cdot N_e \rceil]$ ▷ Threshold at k_1-quantile
$\mathbf{m} \leftarrow \mathbf{1}_{\mathbf{e} \geq f}$ ▷ Initial binary mask
while $\|\mathbf{m}\|_0 > (1 - k_2) \cdot N_e$ **do**
  $i^* \leftarrow \arg\min_i \mathbf{R}_{\text{sum}}(i)$ ▷ Find least used pattern
  Remove column i^* from $\mathbf{D}$ and row i^* from $\mathbf{R}$
  Recompute $\mathbf{R}_{\text{sum}}$, $\mathbf{D}_{\text{sum}}$, $\mathbf{e}$
  Update $\mathbf{m} \leftarrow \mathbf{1}_{\mathbf{e} \geq f}$ ▷ Adapt mask
end while
return $\mathbf{m}$

5.2 Experiments on General and Domain-Specific Tasks

We conduct a series of experiments to evaluate the effectiveness of our proposed pruning method, CAEP. We perform experiments on both general tasks and domain-specific tasks. The goal is to assess how well the pruned model retains its capabilities across a variety of tasks, while optimizing performance retention in specific domains. The dataset and specific configurations used in this part of the experiment can be found in Appendix C.

5.2.1 Experiments on General Tasks

The goal of this experiment is to evaluate how well the pruned model retains its performance across a broad set of general tasks. We compare CAEP with baseline pruning methods to analyze the trade-off between reducing the number of experts and maintaining task performance.

Comparison with Other Expert Pruning Baselines. We compare CAEP to two baseline pruning strategies: (1) Routing Score-Based Pruning (Muzio et al., 2024): Retains experts with higher averaged routing scores. (2) Behavior-based Pruning (Zhang et al., 2024): Remove experts with minimal impact on the output.

Results and Discussion. Figure 8 and Table 1 show that CAEP-pruned models maintain competitive performance, outperforming random and other baseline methods with an average score of 0.650 on DeepSeek and score of 0.652 on Phi-MoE. Notably, CAEP retains higher performance on DeepSeek model after pruning 25% of the experts, especially on tasks like OBQA and RTE. This is further supported by Figure 8, where CAEP shows a low accuracy drop on both DeepSeek and Phi-MoE across multiple tasks even with a high pruning ratio.

Through the analysis of the experimental results, we found that CAEP effectively retains performance across a broad set of general tasks while significantly reducing the number of experts. This demonstrates that **CAEP successfully balances pruning and performance retention, optimizing computational efficiency while minimizing performance loss.**

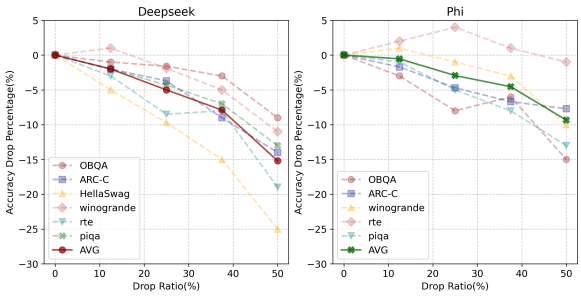

Figure 8: Performance of CAEP on benchmark tasks with varying expert pruning drop ratios.

5.2.2 Experiments on Domain-Specific Tasks

In this experiment, we focus on investigating how expert collaboration patterns differ across

Table 1: Performance evaluation of different expert pruning methods with 25% experts dropped.

Model	Method	AVG↑	OBQA↑	ARC-C↑	HellaSwag↑	WinoGrande↑	RTE↑
DeepSeek	original model	0.692	0.491	0.732	0.791	0.655	0.791
	Random	0.524	0.363	0.564	0.485	0.568	0.641
	SEER-MoE	0.626	0.420	0.672	0.665	0.617	0.755
	GEM	0.628	0.422	0.67	0.658	**0.649**	0.739
	CAEP (Ours)	**0.650**	**0.473**	**0.693**	**0.691**	0.635	**0.757**
Phi-MoE	original model	0.675	0.508	0.534	0.799	0.766	0.769
	Random	0.530	0.410	0.390	0.660	0.580	0.610
	SEER-MoE	0.588	**0.470**	0.450	0.720	0.660	0.640
	GEM	0.636	0.400	**0.530**	0.740	0.720	0.790
	CAEP (Ours)	**0.652**	0.430	0.510	**0.750**	**0.760**	**0.810**

various domains and how these differences reflect the domain-specific interactions within MoE LLMs. Our objective is to analyze the activation frequencies of experts for inputs from five fields—mathematics, computer science, physics, law, and psychology—in order to identify domain-dependent patterns in expert selection. For each domain, we prune 50% of the experts using CAEP on Phi-MoE. This setup enables us to assess whether the pruned model retains superior performance in a specific domain at the expense of others.

Performance Evaluation Metric. To assess the impact of pruning, we focus primarily on the relative changes in performance. The metric is computed as:

$$\frac{Acc_{\text{pruned}} - Acc_{\text{no-pruned}}}{Acc_{\text{no-pruned}}}. \tag{7}$$

A higher value indicates better retention of domain-specific capabilities, with the ideal result being maximized diagonal elements, showing that each pruned model retains domain-specific expertise.

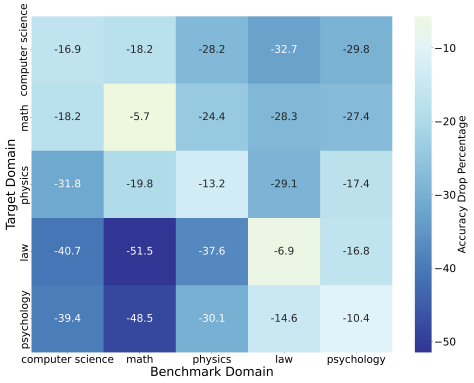

Figure 9: Performance degradation accuracy after pruning for specific domain

Results and Discussion. Figure 9 shows the accuracy degradation after pruning for different domains, presented as a heatmap. The color scale indicates the percentage of accuracy drop, where darker blue shades represent larger losses. From the figure, we observe that pruning for domains like law and psychology leads to the most significant accuracy drops, particularly when the target domain is law. In contrast, pruning for the "physics" or "psychology" domains results in relatively smaller accuracy drops, suggesting a less severe impact on performance.

We find that this variation in pruning impact, depending on both the target and benchmark domains, reveals an uneven distribution of domain-specific knowledge across the model. Some domains rely more heavily on specialized expertise, while others are more flexible in terms of expert collaboration. These findings suggest that **pruning strategies should account for the varying importance of domain-specific knowledge, allowing for more efficient expert retention and minimizing unnecessary performance degradation in MoE LLMs.**

6 Conclusion

This paper addresses a key gap in MoE LLMs, where existing research has largely overlooked the collaboration patterns among experts, both within the same layer and across layers. By applying hierarchical sparse dictionary learning, we uncover dominant expert collaboration patterns and develop a pruning strategy to enhance MoE LLMs' efficiency. Our experiments demonstrate that this approach not only improves accuracy but also significantly boosts model compression and inference efficiency compared to existing methods. This work provides valuable insights into expert interactions and offers a novel way to optimize MoE LLMs for both performance and scalability.

References

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. 2024. A Survey on Mixture of Experts. *arXiv preprint*.

Chen Chen, Hao Su, Qixing Huang, Lin Zhang, and Leonidas Guibas. 2013. Pathlet learning for compressing and planning trajectories. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 392–395, Orlando Florida. ACM.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 2024. DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models. *arXiv preprint*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Leo Gao, Gabriel Goh, and Ilya Sutskever. Scaling and evaluating sparse autoencoders.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, d Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Shwai He, Daize Dong, Liang Ding, and Ang Li. 2024. Demystifying the Compression of Mixture-of-Experts Through a Unified Framework. *arXiv preprint*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Zhi Hou, Xiaojiang Peng, Yu Qiao, and Dacheng Tao. 2020. Visual Compositional Learning for Human-Object Interaction Detection. In *Computer Vision – ECCV 2020*, pages 584–600, Cham. Springer International Publishing.

Zhi Hou, Baosheng Yu, Yu Qiao, Xiaojiang Peng, and Dacheng Tao. 2021. Detecting Human-Object Interaction via Fabricated Compositional Learning. pages 14646–14655.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts. *arXiv preprint*.

Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. 2023. Merge, then compress: Demystify efficient smoe with hints from its routing policy. *CoRR*.

Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. 2024. A Closer Look into Mixture-of-Experts in Large Language Models. *arXiv preprint*.

Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. 2024. Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models. *CoRR*, abs/2402.14800.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 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*, pages 2381–2391.

Alexandre Muzio, Alex Sun, and Churan He. 2024. SEER-MoE: Sparse Expert Efficiency through Regularization for Mixture-of-Experts. *arXiv preprint*.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. 2024. Improving Dictionary Learning with Gated Sparse Autoencoders. *arXiv preprint*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Yuanbo Tang, Zhiyuan Peng, and Yang Li. 2023. Explainable Trajectory Representation through Dictionary Learning. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '23, pages 1–4, New York, NY, USA. Association for Computing Machinery.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Yi Ma. 2009. Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(31):210–227.

Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. 2024. Openmoe: An early effort on open mixture-of-experts language models. In *Forty-first International Conference on Machine Learning*.

Jianchao Yang, John Wright, Thomas S. Huang, and Yi Ma. 2010. Image super-resolution via sparse representation. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 19(11):2861–2873.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Zeliang Zhang, Xiaodong Liu, Hao Cheng, Chenliang Xu, and Jianfeng Gao. 2024. Diversifying the Expert Knowledge for Task-Agnostic Pruning in Sparse Mixture-of-Experts. *arXiv preprint*.

Appendix

A Limitations

The entire work operates under the assumption that the allocation result provided by the router is the most optimal. However, this may only reflect one aspect of the model's behavior. By considering both router information and weight data in a more comprehensive way, we could gain a deeper and more complete understanding. Additionally, there has been limited analysis from the perspective of combinatorial learning, which might offer useful insights into the task selection process. Moreover, the labeling of mined patterns has primarily been done manually up until now, which is very labor-intensive.

B Pruning Effect Calculation

For the DeepSeek-MoE-16B model, considering the significant impact of shared experts on the model, we only prune the normal experts during the pruning operation. Through calculations, we estimate the parameter counts of various parts of DeepSeek-MoE-16B as follows: word embeddings 0.2B, attention mechanism 0.4B, gate and shared experts 0.9B, routing network of MoE 14.7B, and output layer 0.2B. Therefore, for this model, our conclusion is that the total parameters after pruning with a pruning ratio of $k\%$ can be calculated as:

$$\text{New Total Parameters} = (16.4 - 14.7 \times k\%)\,\text{B} \tag{8}$$

C Experiment Setup

C.1 HSDL Experiment Setup

Hierarchical Sparse Dictionary Learning (HSDL) is a critical component of CAEP algorithm. Unlike direct training on the main task, the HSDL module requires additional pre-training. This section aims to elaborate on the experiment setup for the HSDL module, specific training details, and the associated computational overhead.

During the dictionary training phase of HSDL, we selected 604,109 tokens from the MMLU-Pro dataset as training data. The entire dictionary training process took approximately 1200 seconds for 2000 epochs on 4 NVIDIA 4090D GPUs. We set the initial learning rate to 4e-4 and decay the learning rate 0.5 times with every 500 epochs. Despite this additional computational cost, considering the

significant performance improvements brought by the CAEP method in subsequent tasks, we believe this overhead is worthwhile and within an acceptable range.

The core of HSDL lies in its dictionary learning and optimization process. The objective of this process is to minimize a comprehensive loss function, denoted as L_{total} in Equation 7. This total loss function is composed of three key parts: L_{sparse}, L_{hier} and L_{rec}. L_{sparse} represents the sparsity constraint, aiming to ensure that the learned representations are sparse; L_{hier} represents the inter-layer consistency constraint, used to maintain consistency between dictionaries at different hierarchical levels; and L_{rec} represents the reconstruction constraint, ensuring that the input signal can be effectively reconstructed from the sparse representation and the dictionary.

C.2 Pruning Experiment Setup

In section 5, following the setup in (He et al., 2024), we implement our pruning method on the MMLU (Hendrycks et al., 2021) dataset, using 128 samples with an input sequence length of 2,048 tokens. All pruning experiments are conducted on the DeepSeek-MoE-16B model and Phi-MoE model, where only normal experts are pruned, preserving shared experts due to their importance. Model performance is evaluated using the LM-Harness benchmark, which includes a range of tasks: ARC-C (Clark et al., 2018), HellaSwag (Zellers et al., 2019) , OBQA (Mihaylov et al., 2018), RTE (Wang et al., 2019), and WinoGrande (Sakaguchi et al., 2021). The evaluation is carried out using the EleutherAI LM Harness framework (Gao et al., 2023), and we report normalized zero-shot accuracy for each task.

D Comparison of Pruning Algorithms Under 75% Pruning Rate

From Table 2, we can see that our pruning scheme still outperforms the baselines at a pruning ratio of 75%.

E Semantic Annotation for Expert Collaboration Patterns

We also conducted similar analyses on DeepSeek-MoE using MMLUPro as dataset. Here's how the original text was processed by the hierarchical expert collaboration.

	SEER-MoE	GEM	Ours
AVG	0.363	0.387	**0.398**
OBOA	0.252	0.292	**0.298**
ARC-C	0.249	0.278	**0.286**
HellaSwag	0.309	0.356	**0.363**
WinoGrande	0.517	0.504	**0.512**
RTE	0.516	0.538	**0.552**
PIQA	0.337	0.358	**0.380**

Table 2: Comparison of performance between different pruning algorithm across benchmarks when the pruning rate is 75%.

```
---Layer0 - Original Text---
..........|
..........|----Layer1-dict27
..........|.........|
..........|.........|----Layer2-dict659
..........|
..........|----Layer1-dict446
....................|
....................|----Layer2-dict94
....................|
....................|----Layer2-dict1156
```

Original Text: /Q 2: A radioactive material, such as thorium-234, disintegrates at a rate proportional to the amount currently present. If $Q(t)$ is the amount present at time t, then $\frac{dQ}{dt} = -rQ$, where $r > 0$ is the decay rate. If 70 mg of thorium-234 decays to 28 mg in one week, determine the decay rate r.

Relationship between tokens and expert combinations:

First Layer Breakdown:

- Layer1-dict27 Contextual Text: Radioactive material, such as thorium, disintegrates at a rate
- Layer1-dict446 Mathematical Calculation: /Q 2: A rate. If $Q(t)$, t, then $\frac{dQ}{dt} = -r$, $r > 0$ 70

Second Layer Breakdown:

- Layer2-dict659 Contextual Text: Radioactive material, such as thorium, disintegrates at a rate
- Layer2-dict94 Numbers: 2 34 0 0 70

- Layer2-dict1156 Symbols: /Q : -İf $Q(t)$ is t, $\frac{dQ}{dt} = -r$

From the above examples, it can be observed that in different MOE models, we can also find various expert collaboration patterns with clear tendencies. Furthermore, based on our extensive experiments with other MOE models and datasets, the aforementioned patterns are widely present, indicating that our method possesses strong generality.