Specialization through Collaboration: Understanding Expert Interaction in Mixture-of-Expert Large Language Models

Anonymous ACL submission

Abstract

Mixture-of-Experts (MoE) based large language models (LLMs) have gained popularity due to their multi-task capability, where each input token activates only a subset of "expert" subnetworks. However, whether each expert can truly specialize to a certain task remains poorly understood, while activation analysis shows frequent cross-layer co-activation of experts for the same input, resembling a collaborative behavior. In this paper, we use a dictionary learning approach to show that experts in MoE LLMs form hierarchical and semantically coherent collaborative groups that correspond to specific linguistic and cognitive functions (e.g., mathematical reasoning, syntactic processing), mirroring specialized functional region observed in neuroscience. Furthermore, leveraging these discovered expert groups enables significant model compression with minimal performance degradation, outperforming existing methods by 2.5% while enabling up to 50% expert reduction. These findings provide the first systematic analysis of expert collaboration mechanisms in MoE LLMs, revealing that specialization emerges from joint activation of experts across all layers. We further developed an interactive visualization platform that enables researchers to explore expert collaboration patterns and their semantic associations. The code repository is available at this URL.

1 Introduction

Mixture-of-Experts (MoE) based large language models have emerged as a promising paradigm for handling diverse tasks, where different expert subnetworks are selectively activated for each input token (Jiang et al., 2024; Fedus et al., 2022). Operating as an implicit multi-task learning framework without requiring explicit task boundaries, MoE architectures demonstrate remarkable adaptability across various domains (Cai et al., 2024). This flexibility has positioned MoE LLMs as a cornerstone for scalable and efficient language modeling in contemporary AI systems.

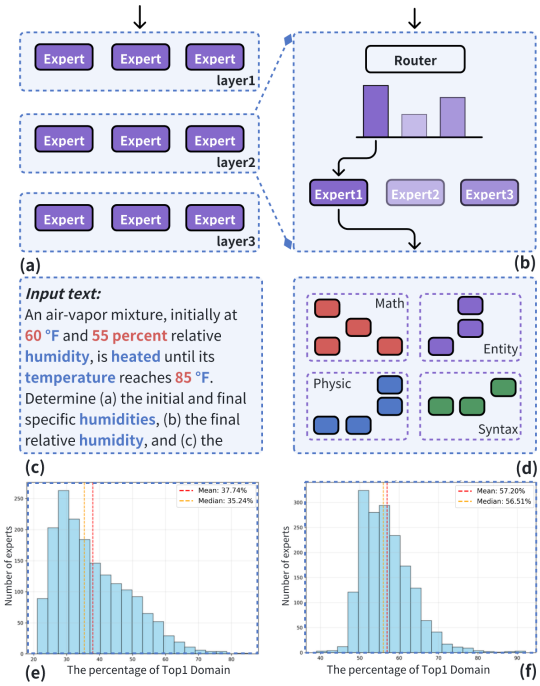

Figure 1: (a) Multi-layer MoE structure; (b) Routing and expert selection; (c) Token-to-expert mapping; (d) Domain-specific expert combinations; (e) Distribution of the percentage of Top1 Domain activated by individual experts. (f) Distribution of the percentage of Top1 Domain activated by expert groups.

However, a critical question remains unresolved: do individual experts in MoE LLMs genuinely specialize in distinct tasks? Previous studies on activation pattern analysis have shown that topic-based expert assignment patterns are not readily observable (Jiang et al., 2024). Inspired by neuroscience research demonstrating that neural populations rather than individual neurons are responsible for basic functions (Panzeri et al., 2022), this leads us to consider whether specialization in MoE networks similarly emerges from coordinated interac-

tions. Furthermore, neuroscience research has revealed that higher-order correlations among neural populations create complex organizational structures (Panzeri et al., 2022), motivating our investigation into hierarchical expert collaboration patterns in MoE architectures.

In this paper, we propose a dictionary learning framework to analyze expert collaboration patterns. Unlike previous approaches that examine individual expert behaviors or router decisions (Lo et al., 2024; Jiang et al., 2024), our method identifies coordinated expert groups that activate jointly across layers. We validate our approach through comprehensive experiments on representative MoE architectures using the MMLU-pro benchmark, encompassing 2,812 samples across mathematics, computer science, physics, law, and psychology domains. The analysis uncovers hierarchical collaboration structures where expert combinations correspond to specific linguistic and cognitive functions, such as mathematical reasoning or syntactic processing, analogous to specialized functional regions observed in neuroscience (Panzeri et al., 2022). As shown in subplots (e,f) of Figure 1, expert groups are more focused on processing tokens from specific domains compared to individual experts. These discovered patterns provide the first systematic characterization of how specialization emerges in MoE LLMs from an expert group perspective.

Building upon these discoveries, we demonstrate practical applications of collaboration pattern analysis for model compression. By identifying which expert groups contribute most significantly to specific tasks, we develop a pruning strategy that selectively removes low-contribution experts while preserving essential collaborative structures. Our experimental results show substantial parameter reduction (up to 50% of experts in certain configurations) with minimal performance degradation, outperforming existing pruning methods by 2.5% on average. This effectiveness serves as empirical validation that the identified collaboration patterns capture semantic structures within MoE architectures.

The primary contributions of this work are:

- We present a comprehensive analysis of cross-layer expert collaboration in MoE LLMs, revealing hierarchical group structures where coordinated expert combinations implement specific cognitive functions.
- We demonstrate practical implications of collaboration pattern discovery for model optimization, achieving superior compression performance by preserving functionally critical expert groups while removing redundant components.
- We develop an interactive visualization platform that enables real-time exploration of expert activation dynamics and collaboration patterns, facilitating interpretability research and model diagnostics for MoE architectures.

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 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,

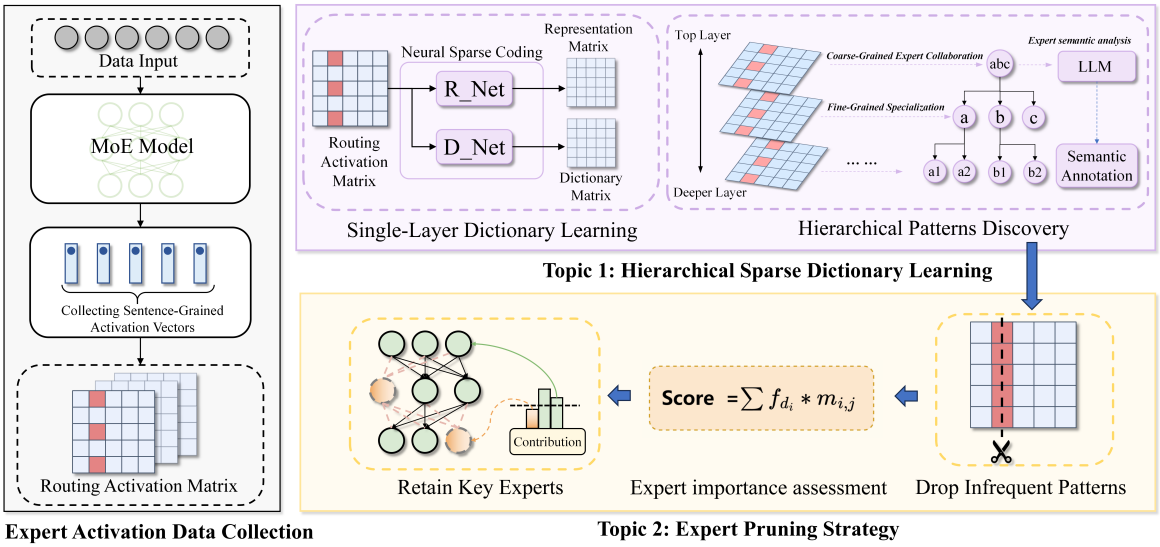

Figure 2: Overview of Our Study's Framework.

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

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:

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

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

Co-activation between experts is common in MoE networks. For instance, Figure 3 presents a concrete example of expert collaboration in DeepSeek-MoE: Expert 21 in Layer 5 and Expert 3 in Layer 6 demonstrate strong co-activation tendencies. We propose a novel **Hierarchical Sparse Dictionary Learning (HSDL)** approach that captures coordinated expert groups across layers through multi-level decomposition. We validate this framework on the MMLU-pro dataset, demonstrating how discovered patterns correspond to specific cognitive functions and reveal domain-specific expert interactions.

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 introducing hierarchical sparse dictionary learning (HSDL) 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 $\mathbf{X} \approx \mathbf{D} \cdot \mathbf{R}$.

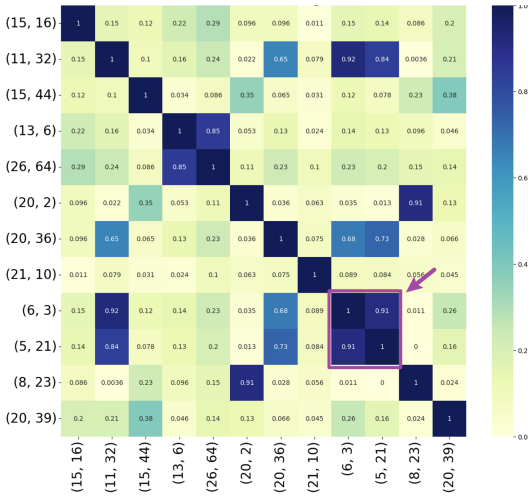

Figure 3: Here (x, y) refers to the y-th expert in x-th layer. By selecting any two experts from the MoE, we can calculate the probability of their co-activation. It can be observed that Expert 21 from the layer 5 and Expert 3 from the layer6 frequently activate simultaneously, forming an expert collaboration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

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.

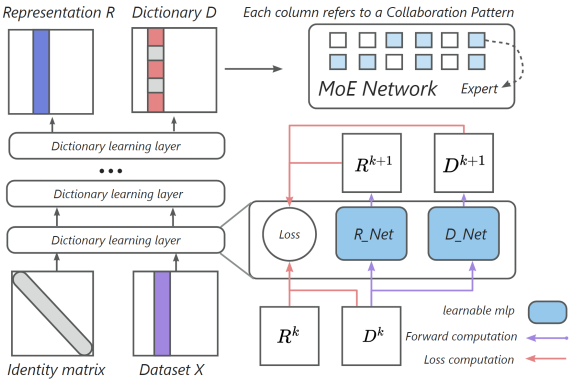

Figure 4: Hierarchical Sparse Dictionary Learning.

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 limitation, we propose the Hierarchical Sparse Dictionary Learning (HSDL) approach, which extends the single-layer dictionary learning framework from (Zhao et al., 2025) to a multi-layer hierarchical structure. This hierarchical extension enables the capture of collaboration patterns from coarse to fine granularity, thus revealing hierarchical expert interactions that cannot be detected by traditional single-layer methods. Specifically, we recursively decompose 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.

To optimize this hierarchical structure, we referenced three key constraints from (Zhao et al., 2025), a Bayesian-based dictionary learning method, extending their framework into a hierarchical multi-layer version:

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

$$L_{\text{Rec}} = \sum_{k=0}^{K} \sum_{j=0}^{M} \|D^k_{:,j} - (D^{k+1} R^{k+1})_{:,j}\|_1 \cdot \frac{\|R^k_{:,j}\|_1}{N}. \tag{2}$$

(2) **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:

$$L_{\text{Dict}} = \sum_{k=0}^{K} \sum_{i=0}^{M} \|R^k_{i,:}\|_\infty. \tag{3}$$

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

$$L_{\text{Sparse}} = \sum_{k=0}^{K} \sum_{j=0}^{M} \|R^{k+1}_{:,j}\|_1 \cdot \|R^k_{:,j}\|_1 / N. \tag{4}$$

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

$$L_{\text{Total}} = L_{\text{Rec}} + \lambda_1 L_{\text{Dict}} + \lambda_2 L_{\text{Sparse}}, \tag{5}$$

where λ_1 and λ_2 are hyperparameters that control the respective losses. In our implementation, we set

the hierarchical depth $K = 2$ to capture two levels of expert collaboration patterns, and use $\lambda_1 = 1$ and $\lambda_2 = 1$ to balance the three loss components equally. The values of λ_1 and λ_2 can be computed from the training data following the method described in (Zhao et al., 2025). 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 Semantic Annotation Method for Expert Collaboration Pattern Interpretation

To interpret the semantic meaning of expert collaboration patterns discovered by HSDL, we develop a semantic annotation method consisting of two steps: (1) mapping dictionary atoms to input tokens based on activation strength in the sparse coding matrix R^k, and (2) employing large language models to automatically generate semantic descriptions for each pattern based on the tokens it processes. The specific prompt for automatic annotation is provided in F.

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

4.4.1 Hierarchical Semantic Analysis of Expert Collaboration Patterns

To explore how expert collaboration patterns in MoE LLMs reflect the model's understanding of tasks, we apply our semantic annotation method to analyze input samples using the dictionary atoms obtained through HSDL. We visualize words processed by the same dictionary atoms (i.e., expert collaboration patterns) with the same color to facilitate the observation of interrelationships. One such analysis is 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. 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 Figure 5 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 E for more examples).

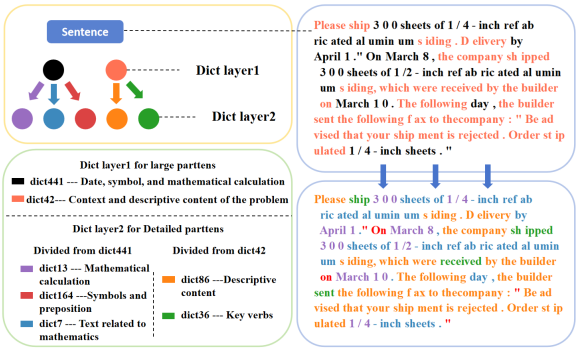

Figure 5: Hierarchical Semantic Annotation of Dictionary Elements on MMLU.

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. Additionally, we developed an interactive visualization platform as demonstrated in Figure 6.

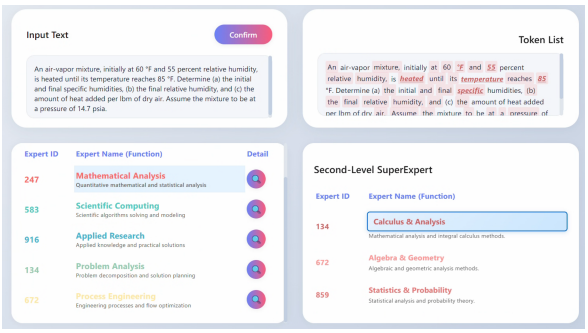

Figure 6: Interactive website interface for analyzing hierarchical correspondence between MoE expert groups and tokens.

4.4.2 Domain-Specific Expert Collaboration Patterns

In this experiment, our goal is to explore how expert collaboration patterns vary across different do-

mains 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.

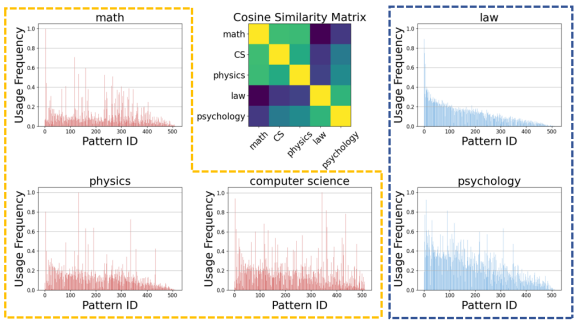

Figure 7: The distribution of expert selection frequencies during inputs from different fields.

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 B.

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. We also experimented with different combinations of loss objectives to evaluate the contribution of each component, and found that the absence of either L_{sparse} or L_{dict} leads to performance degradation.

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.**

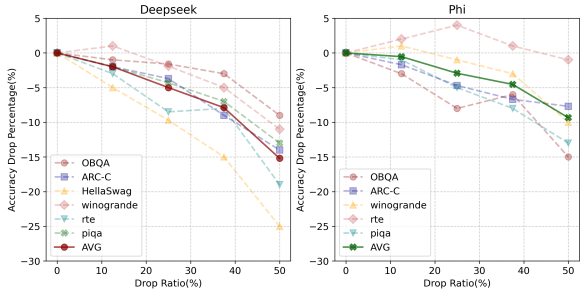

Figure 8: Performance of CAEP on benchmark tasks with varying expert pruning drop ratios.

5.2.2 Experiments on Domain-Specific Tasks

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

$$\frac{Acc_{\text{pruned}} - Acc_{\text{no-pruned}}}{Acc_{\text{no-pruned}}}. \tag{6}$$

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.

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.

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**
	CAEP ($L_{rec} + L_{sparse}$)	0.641	0.461	0.698	0.667	0.628	0.751
	CAEP ($L_{rec} + L_{dict}$)	0.645	0.470	0.681	0.682	0.631	0.761
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**
	CAEP ($L_{rec} + L_{sparse}$)	0.636	0.418	0.495	0.732	0.745	0.792
	CAEP ($L_{rec} + L_{dict}$)	0.643	0.423	0.502	0.738	0.751	0.798

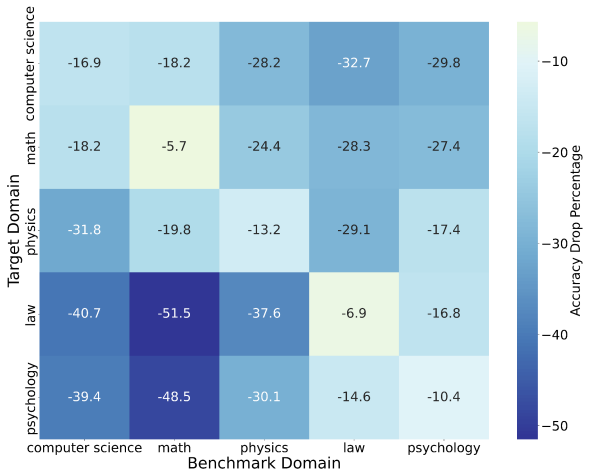

Figure 9: Performance degradation accuracy after pruning for specific domain

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.

Limitations

Validation on Different MoE Architectures. Our study has only validated the proposed method on two pre-trained MoE models (DeepSeekMoE and Phi). While these models demonstrate the effectiveness of our approach, the generalizability to a broader range of MoE architectures remains to be established. Additionally, we have not extensively explored how different architectural designs influence the relationship between expert collaboration and specialization patterns. Understanding these dependencies would be valuable for designing future MoE architectures.

Dictionary Learning Method Limitations. Our approach has not extensively compared with other pattern mining methods, and we have not performed extensive hyperparameter optimization for the dictionary learning framework. However, our focus is on discovering expert collaboration and specialization relationships rather than optimizing pattern mining techniques. Dictionary learning was chosen due to its proven effectiveness in pattern mining and data compression (Yang et al., 2010; Wright et al., 2009).

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*.

Stefano Panzeri, Monica Moroni, Houman Safaai, and Christopher D. Harvey. 2022. The structures and functions of correlations in neural population codes. *Nature Reviews Neuroscience*, 23(9):551–567.

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*.

Zihui Zhao, Yuanbo Tang, Jieyu Ren, Xiaoping Zhang, and Yang Li. 2025. A Unified Probabilistic Framework for Dictionary Learning with Parsimonious Activation. *arXiv preprint*. ArXiv:2509.25690 [cs].

Appendix

A 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:

$$\text{New Total Parameters} = (16.4 - 14.7 \times k\%)\,\text{B} \tag{7}$$

B Experiment Setup

B.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.

B.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.

C 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%.

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

	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**

D 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.

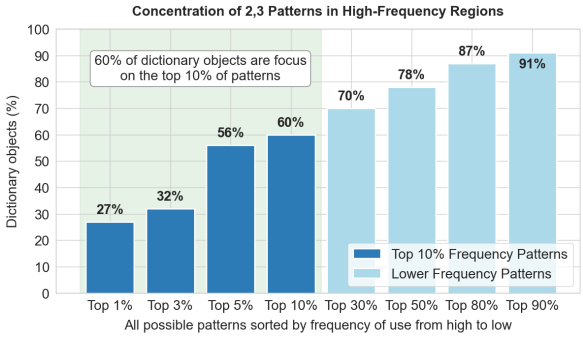

Figure 10: Comparison of overlap with the results of the exhaustive method.

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 calculated using the following formula:

$$\text{Top } k\% \text{ Coverage} = \frac{N_{\text{top}}}{N_{\text{total}}}. \tag{8}$$

Results and Discussion. As shown in Figure 10, **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.**

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.

Layer Breakdown

```
—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: Radio active 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.

F LLM Annotation Prompt

The following prompt was used for automatic annotation of expert functional descriptions by large language models:

LLM Annotation Prompt

```
prompt = f"""Your job is to look for
patterns in text. You will be given a list
of WORDS, your task is to provide an
explanation for what pattern best
describes them. Here are some guidelines:
- Produce a specific final description for
the latents common in the examples, and
what patterns you found.
- Don't focus on giving examples of
important tokens, if the examples are
uninformative, you don't need to mention
them.
- Do not make lists of possible
explanations. Keep your explanations
short and concise.
- The last line of your response must be
the formatted explanation, using
[EXPLANATION]:
WORDS: {token_list}
"""
```

G Computational Cost Analysis of HSDL Method

This section provides a detailed quantitative analysis of the computational costs and benefits associated with our Hierarchical Sparse Dictionary Learning (HSDL) method compared to baseline approaches.

G.1 Direct Comparison of Computational Costs

G.1.1 Cost of HSDL Method

The computational cost of our HSDL method consists of two main components: expert activation data collection and dictionary training.

Activation Data Collection: This requires a single sparse forward pass on the calibration dataset. Using the standard FLOPs estimation for an MoE Transformer pass: FLOPs $\approx 2 \times P_{\text{activated}} \times L_{\text{total}}$.

For DeepSeek-MoE:

- Activated parameters per token: $P_{\text{activated}} = 2.8 \times 10^9$
- Total tokens from MMLU-Pro dataset: $L_{\text{total}} = 604,109$
- Collection cost: $2 \times (2.8 \times 10^9) \times 604,109 \approx 3380$ TFLOPs

HSDL Dictionary Training: The training cost is calculated using $\text{FLOPs}_{\text{Training}} \approx 6 \times P \times D_{\text{total}}$, where the factor of 6 accounts for forward and backward passes.

- Total optimized parameters: $P = (1728 \times 1728) + (1728 \times 800) \approx 4.37 \times 10^6$

- Training steps: $D_{\text{total}} = 2000$ epochs
- Training cost: $6 \times (4.37 \times 10^6) \times 2000 \approx 0.052$ TFLOPs

Total HSDL cost: $3380 + 0.052 \approx 3380$ TFLOPs.

G.1.2 Cost of Baseline Methods

Baseline methods (SEER-MoE, GEM) require at least one full forward pass over a calibration dataset to calculate importance scores:

- Processing approximately 65,536 tokens (32 sequences of 2048 tokens)
- For the 16.4B DeepSeek-MoE model: $2 \times (16.4 \times 10^9) \times 65,536 \approx 2150$ TFLOPs

Table 3: Computational cost comparison of different methods.

	HSDL (Ours)	**Baseline Methods**
Cost	$\sim$ 3380 TFLOPs	$\sim$2150 TFLOPs
Cost Type	Sparse Pass + Offline Training	Full-Model Forward Pass

G.2 Quantifying the Benefits of Pruning

The modest one-time cost of HSDL enables significant permanent benefits through CAEP pruning. For the DeepSeek-MoE-16B model with a 25% pruning ratio:

- **Parameter Reduction:** Permanent reduction of 3.675 billion parameters
- **Storage & Memory Savings:** Approximately 14.7 GB permanent savings (float32 precision)

These savings are critical for making large MoE models more practical for deployment and serving, directly reducing hardware requirements and operational costs.

G.3 Trade-off Analysis

For a comparable one-time computational cost ($\sim$3380 TFLOPs vs. $\sim$2150 TFLOPs for baselines), our method invests this budget in a more sophisticated analysis. While baselines perform simple forward passes to collect individual expert metrics (like frequency), our method first gathers rich activation data and then executes extremely efficient optimization (HSDL) to uncover complex, hierarchical collaboration patterns.

This "smarter" investment pays off significantly in the pruning stage. As demonstrated in Table 1, our CAEP method (0.650 AVG score) delivers superior performance retention compared to the baselines (GEM: 0.628, SEER-MoE: 0.626). This proves that the patterns captured by our analysis are more effective for identifying crucial experts than the simpler metrics used by other methods.

In conclusion, for a similar one-time computational price, our method yields a more effective pruning strategy that leads to significant permanent memory savings while better preserving model performance. We are not just pruning; we are investing in understanding the model to prune intelligently.